2007

Stability and Static Noise Margin Analysis of Static Random Access Memory

Rajasekhar Keerthi
Wright State University

STABILITY AND STATIC NOISE MARGIN ANALYSIS OF STATIC RANDOM ACCESS MEMORY

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering

By

RAJASEKHAR KEERTHI
B.E., University of Madras, 2004

2007
Wright State University

WRIGHT STATE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

November 20, 2007

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Rajasekhar Keerthi ENTITLED Stability and Static-Noise-Margin Analysis of Static Random Access Memory BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Engineering.

_____

Chien-In Henry Chen, Ph.D.
Thesis Director

_____

Fred Garber, Ph.D.
Department Chair

Committee on
Final Examination

_____

Chien-In Henry Chen, Ph.D.

_____

Raymond Siferd, Ph.D

_____

Marian K. Kazimierczuk, Ph.D.

_____

Joseph F. Thomas, Jr., Ph.D.
Dean of the School of Graduate Studies

Abstract

Keerthi, Rajasekhar M.S.E., Department of Electrical Engineering, Wright State University, 2007, *Stability and Static Noise Margin Analysis of Static Random Access Memory*

The transistor mismatch can be described as two closely placed identical transistors have important differences in their electrical parameters as threshold voltage, body factor and current factor and make integrated circuit design and fabrication less predictable and controllable. Stability of a static random access memory (SRAM) is defined through its ability to retain the data at low-V_{DD}. It is seriously affected by increased variability of transistor mismatch and decreased supply voltage and therefore becomes a major limitation of overall performance of low-voltage SRAM in nanometer CMOS process. The stability limitation is addressed through the design of a seven-transistor (7T) SRAM cell and of which the stability analysis and comparison with the conventional 6T SRAM cell is presented. This research also presents two 8-bit SRAM designs implemented by 6T and 7T SRAM cells respectively. The robustness of both designs is tested and verified through transistor mismatch and environmental process variations. Results obtained show 7T SRAM outperform 6T SRAM when stability is of a major concern.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to express gratitude to my advisor, Dr. Chien-In Henry Chen, for his invaluable support, encouragement, supervision and useful suggestions throughout this research work.

I wish to thank Dr. Raymond E. Siferd and Dr. Marian K. Kazimierczuk for serving as members of my MS thesis defense committee, giving the constructive criticism and comments. I would also like to thank Mr. Kumar Yelamarthi for his valuable suggestions throughout this thesis work.

I wish to thank the Department of Electrical Engineering and Dr. Fred D. Garber, the Department Chair, for giving me the opportunity to obtain my MS degree at Wright State University.

Finally, I would like to thank my family and friends for their constant encouragement and support.

Dedicated to my Family

Introduction to Memory Array

1.1 Introduction

SRAM is a critical component in many of the digital systems, from high-performance processors to mobile-phone chips. In these applications, density, power, and performance are all essential parameters. Earlier, the power for digital logic, which is dominated by dynamic power, has been reduced by lowering the supply voltage (V_{DD}) [3]. The supply voltage for digital circuits has reached around 1 V [4]–[6]. V_{DD} scaling down reduces static-noise-margin but increases the transistor mismatch [1]. Besides, there are severe constraints on cell noise margin for reliable read-and-write operation. Also, as device size is scaled, random process variations significantly degrade the noise margin. As the sizing of the SRAM is in nanometer scale the variations in electrical parameters (e.g., threshold voltage, sheet resistance) reduces its steadily due to the fluctuations in process parameters i.e., density of impurity concentration, oxide thickness and diffusion depths [1]. Considering all these effects, the bit yield for SRAM is strongly influenced by V_{DD}, threshold voltage (V_{th}), and transistor-sizing ratios [7]. Therefore, it is complicated to determine the optimal cell design for SRAM.

The transistor mismatch can be described as two closely placed identical transistors have important differences in their electrical parameters as threshold voltage (V_{th}) [1], body factor and current factor and make the design with less predictable and controllable. The stability of the SRAM cell is seriously affected by the increase in variability and decrease in supply voltage (V_{DD}) [8].

1.2 Memory Array

The memory array is classified into 3 types - Random Access memory (RAM), Serial access memory and content addressable memory (CAM). We will discuss each type in detail.

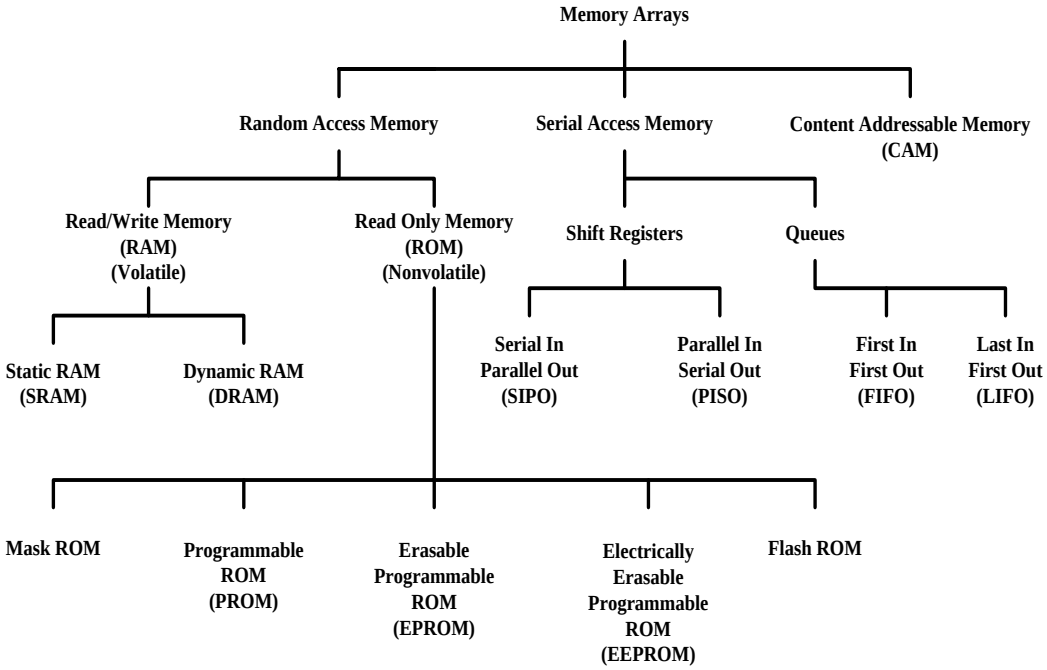

Figure 1.1 Memory Array

1.2.1 Random Access memory (RAM)

Random access memory [17] is a type of computer data storage. It is made of integrated circuits that allow the stored data to be accessed in any order i.e., at random and without the physical movement of storage medium or a physical reading head. RAM is a volatile memory as the information or the instructions stored in the memory will be lost if the power is switched off.

The word “random” refers to the fact that any piece of data can be returned at a constant time regardless of its physical location and whether or not it is related to the previous piece of data. This contrasts with the physical movement devices such as tapes, magnetic disks and optical disks, which rely on physical movement of the recording medium or reading head. In these devices, the retrieval time varies with the physical location and the movement time takes longer than the data transfer.

The main advantages of RAM over types of storage which require physical movement is that retrieval times are short and consistent. Short because no physical movement is necessary and consistent the time taken to retrieve the data does not depend on the current distance from a physical head. The access time for retrieving any piece of data in RAM chip is same. The disadvantages are its cost compared to the physical moving media and loss of data when power is turned off.

RAM is used as 'main memory' or primary storage because of its speed and consistency. The working area used for loading, displaying and manipulating applications and data. In most personal computers, the RAM is not an integral part of the motherboard or CPU. It comes in the easily upgraded form of modules called memory sticks. These can quickly be removed and replaced when they are damaged or when the system needs up gradation of memory depending on current purposes. A smaller amount of random-access memory is also integrated with the CPU, but this is usually referred to as "cache" memory, rather than RAM. Modern RAM generally stores a bit of data as either a charge in a capacitor, as in dynamic RAM, or the state of a flip-flop, as in static RAM.

1.2.1.1 Static Random Access Memory (SRAM)

The word “static” means that the memory retains its contents as long as the power is turned on. Random access means that locations in the memory can be written to or read from in any order, regardless of the memory location that was last accessed. Each bit in an SRAM is stored on four transistors that form two cross-coupled inverters. This storage cell has two stable states which are used to denote ‘0’ and ‘1’. The access transistors are used to access the stored bits in the SRAM during read or write mode.

It thus typically takes six MOSFETs to store one memory bit. Access to the cell is enabled by the word line WL which controls the two access transistors N_1 and N_2 which, in turn, control whether the cell should be connected to the bitlines BL and /BL. They are used to transfer data for both read and write operations. The bitlines are complementary as it improves the noise margin. Chapter 2 explains more about SRAMs and its Read/Write operations.

1.2.1.2 Dynamic RAM

Dynamic Random Access Memory [16] is a type of RAM that stores each bit of data in a separate capacitor within an integrated circuit. As the capacitors leak charge the circuit needs to be refreshed periodically in order to store the data. The structural simplicity of DRAM as it needs one transistor and one capacitor are required per bit where as for SRAM it needs six transistors. This criterion allows DRAM to go very high density. It comes in RAM section as it is volatile; it loses data when the power is turned off.

The periodic refresh operation consists of a read of the cell contents followed by write operation, should occur often enough that the contents of the memory cells are never corrupted by the leakage. Typically refresh should occur every 1 to 4 ms. For a larger memories refresh equipment is placed in the circuit which refreshes every row of the circuit.

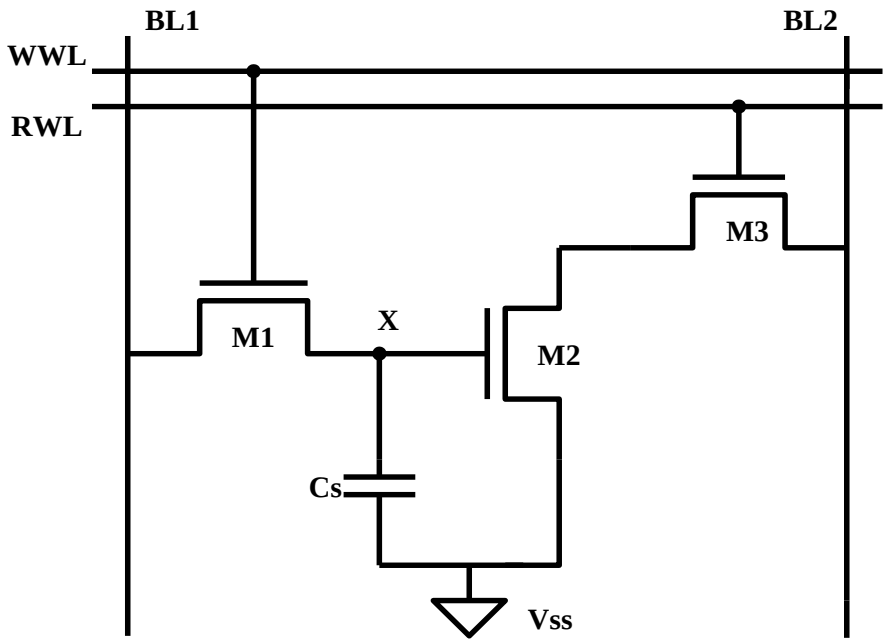

Figure 1.2 Three Transistor Dynamic RAM [Ref. 16]

The write operation performed is shown for three transistor Dynamic RAM (Figure 1.2) as the appropriate data value is written on BL1 and asserting the write-wordline (WWL). The data is retained as charge on capacitance Cs once WWL is lowered. When reading the cell, the read-wordline (RWL) is raised. The storage transistor M_2 is either on or off depending upon the stored value. The bitline BL2 is precharged to V_{DD} before performing read operation. The series connection of M_2 and M_3 pulls BL2 low when a '1'

is stored. BL2 remains high in the opposite case. The cell is inverting; that is, the inverse value of the stored signal is sensed on the bitline.

1.2.2 Read only memory (ROM)

The basic idea of the memory that can only be read and never altered is called Read only memories [16]. There are vast and variety of potential applications for these kind of memories. Programs for processors with fixed applications such as washing machines, calculators and game machines, once developed and debugged, need only reading. Fixing the contents at manufacturing time leads to small and fast implementation.

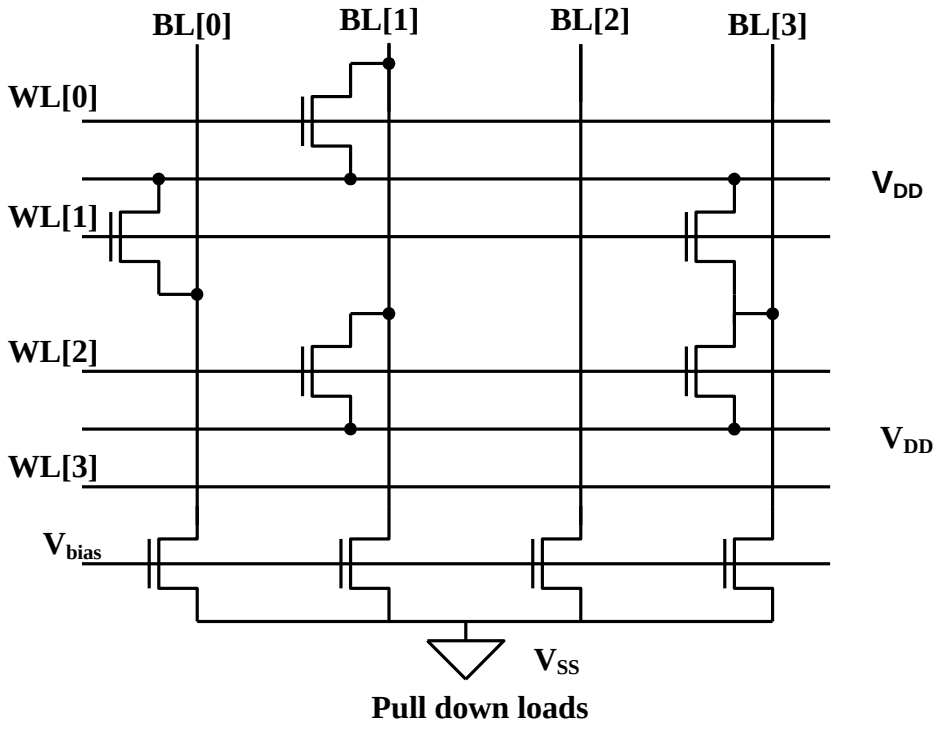

Figure 1.3 A 4 X 4 OR ROM Cell array [Ref. 16].

There are different ways to implement the logic of ROM cells, the fact that the contents of a ROM cell are permanently fixed considerably simplifies its design. The cell should be designed so that a ‘0’ or ‘1’ is presented to the bitline upon activation of its

wordline. The different approaches for implementing the ROM cells are Diode ROM, MOS ROM 1 and MOS ROM 2. These are the main approaches for designing a larger density ROMs.

1.2.2.1 Mask ROM

The ROM memories [16] which we have seen earlier are application specific ROMs where the memory module is part of a larger custom design and programmed for that particular application only. The ROMs which we are going to discuss in this section are commodity ROMs, where a vendor mass-produces memory modules that are later customized according to customer specifications. Under these circumstances, it is essential that the number of process steps involved in programming be minimal and that they can be performed as a last phase of the manufacturing process. In this way large amounts of unprogrammed dies can be preprocessed.

This mask-programmable approach preferably uses the contact mask to personalize or program the memory. The programming of a ROM module involves the manufacturer, which introduces an unwelcome delay in product development. The major usage of this ROM was in system-on-a-chip where the majority of the chip is preprocessed, only the minor part of the die is mask programmed. The other usages of this ROM are to program the microcontroller, embedded on the chip, for a variety of applications.

1.2.2.2 Programmable ROM (PROM)

The technology that offers its users to program the memory one time is called Programmable ROM [16]. It is also called as WRITE ONCE device. This is most often accomplished by introducing fuses (implemented in nichrome, polysilicon, or other

conductors) in the memory cell. During the programming phase, some of these fuses are blown by applying a high current, which disables the connected transistor.

While PROMs have the advantage of being "customer programmable," the single write phase makes them unattractive. For instance, a single error in the programming process or application makes the device unstable. This explains the current preference for devices that can be programmed several times.

The Floating-Gate transistor is the device at the heart of the majority of reprogrammable memories. Various attempts have made to create a device with electrically alterable characteristics and enough reliability to support a multitude of write cycles. The floating gate structure is similar to a traditional MOS device, except that an extra polysilicon strip is inserted between the gate and channel. This strip is not connected to anything and is called a floating gate. The most obvious impact of inserting this extra gate is to double the gate oxide thickness $t_{ox,}$ which results in a reduced device transconductance as well as an increased threshold voltage. Though these properties are not desirable but from other point of view this device acts as a normal transistor.

The most important property of this device is that the threshold voltage of this device is programmable. By applying a high voltage (above 10V) between the source and the gate-drain terminals creates a high electric field and causes avalanche injection to occur. Electrons acquire sufficient energy to become "hot" and traverse through the first oxide insulator, so that they get trapped on the floating gate. In reference to the programming mechanism, the floating-gate transistor is often called a floating-gate avalanche-injection MOS.

The trapping of electrons on the floating gate effectively drops the voltage on the gate. This process is self-limiting – the negative charge accumulated on the floating gate reduces the electrical field over the oxide so that ultimately it becomes incapable of accelerating any more hot electrons. Virtually all nonvolatile memories are currently based on the floating-gate mechanism. Different classes can be identified, based on the erasure mechanism.

1.2.2.3 Erasable-programmable Read-Only Memory (EPROM)

The erasure mechanism in EPROM [16] is based on the shining ultraviolet light on the cells through a transparent window in the package. The UV radiation renders the oxide to conduct by the direct generation of electron-hole pairs in the material. The erasure process is slow depending on the UV source, it can take from seconds to several minutes. The programming takes several μs/word. Alternatively there is another problem which exists is the limited endurance - the number of erase/program cycles is limited to a maximum of one thousand mainly as a result of UV erasing procedure. The device thresholds might vary with repeated programming cycles. The on-chip circuitry is designed in such a way that it also controls the value of the thresholds to within a specified range during programming. The injection of large channel current of 0.5 mA at a control gate voltage of 12.5V causes high power dissipation during programming.

On the other hand, EPROM is extremely simple and dense, making it possible to fabricate large memories at a low cost. Therefore EPROMs were attractive in applications that do not require reprogramming. The major disadvantage of the EPROM is that the erasure procedure has to occur “off system”. This means the memory must be removed from the board and placed in an EPROM programmer for programming.

1.2.2.4 Electrically Erasable Programmable Read-Only Memory (EEPROM)

The disadvantage of the EPROM [16] is solved by using a method to inject or remove charges from a floating-gate namely – tunneling. A modified floating-gate device called FLOTOX (floating-gate tunneling oxide) transistor is used as programmable device that supports an electrical-erasure procedure. It resembles FAMOS (floating-gate avalanche MOS) device, except that a portion of the dielectric separating the floating gate from the channel and drain is reduced in thickness to about 10 nm or less.

The main advantage of this programming approach is that it is reversible; that is, erasing is simply achieved by reversing the voltage applied during the writing process. The electrons injection on floating-gate raises the threshold, while the reverse operation lowers the V_T. When a voltage of approximately 10V (equivalent to 10^9 V/m) is applied over the thin insulator, electrons travel to and from the floating gate through a mechanism called Fowler – Nordheim tunneling.

1.2.2.5 Flash Electrically Erasable Programmable ROM (Flash)

The concept of Flash EEPROMs [16] is a combination of density of EPROM with versatility of EEPROM structures, with cost and functionality ranging from somewhere between two. Most Flash EEPROM devices use the avalanche hot-electron-injection approach to program the device. Erasure is performed using Fowler – Nordheim tunneling, as from EEPROM cells. The main difference is that erasure procedure is performed in bulk for a complete chip or for the subsection of the memory. Erasing complete memory core at once makes it possible to carefully monitor of the device characteristics during erasure.

The monitoring control hardware on the memory chip regularly checks the value of the threshold during erasure, and adjusts the erasure time dynamically. This approach is only practical when erasing large chunks of memory at a time; hence the flash concept. One of the many existing alternatives for Flash EEPROMs memories are ETOX devices. It resembles a FAMOS gate except that a very thin tunneling gate oxide is utilized (10 nm). Different areas of the gate oxide are used for programming and erasure. Programming is performed by applying a high voltage (12V) on the gate and drain terminals for a grounded source, while erasure occurs with the gate rounded and the source at 12V.

The Programming cycle starts with an erase operation. In erase operation, A 0V gate voltage is applied and a 12V supply is given at source. Electrons, if any, are ejected to the source by tunneling. All cells are erased simultaneously. The variations caused in the threshold voltage at the end of erase operation are due to different initial values of cell threshold voltage and variations in oxide thickness. This can be solved in two methods:

1. The array cells are programmed before applying the erase pulse so that the entire threshold starts at approximately same time.

2. An erase pulse of controlled width is applied. Subsequently the whole array is read to ensure that all the cells are erased. If not another erase pulse is applied followed by the read cycle.

For write (programming) operation, a high voltage is applied to the gate of the selected device. If a ‘1’ is applied to the drain at that time, hot electrons are generated and

injected onto the floating gate, raising the threshold. Read operation corresponds as the wordline is raised to 5V; it causes a conditional discharge of bitline.

1.2.3 Serial Memories

Unlike RAMs which are randomly write the data, serial memories restrict the order of access, which results in either faster access times, smaller area, or a memory with a special functionality.

1.2.3.1 Shift Registers

Shift registers are a type of sequential logic circuit, mainly for storage of digital data. They are a group of flip-flops connected in a chain so that the output from one flip-flop becomes the input of the next flip-flop. Most of the registers possess no characteristic internal sequence of states. All the flip-flops are driven by a common clock, and all are set or reset simultaneously. There are two types of shift registers; Serial-in-parallel-out and Parallel-in-serial-out.

Serial-In-Parallel-Out: In this kind of register, data bits are entered serially. The difference is the way in which the data bits are taken out of the register. Once the data are stored, each bit appears on its respective output line, and all bits are available simultaneously. A construction of a four-bit serial in - parallel out register is shown below.

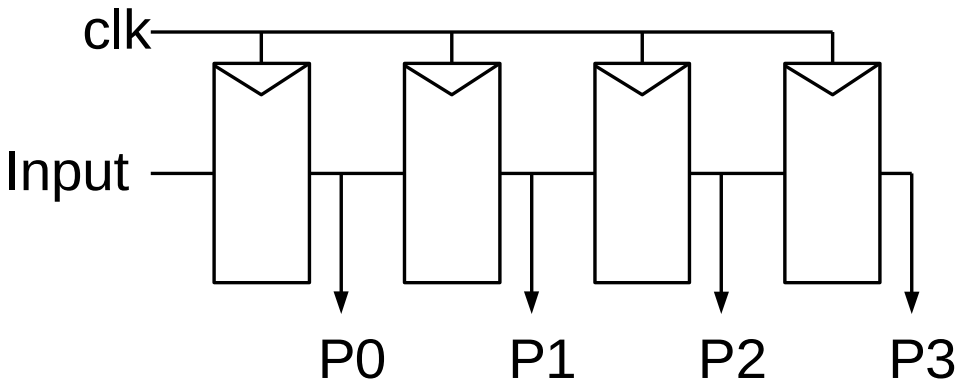

Figure 1.4 Serial-in-parallel-out Shift Register

Parallel-In-Serial-Out: The figure shown below is an example of Parallel-In-Serial-Out shift register. P0, P1, P2 and P3 are the parallel inputs to the shift register. When Shift = '0' the shift register loads all the inputs. When Shift = '1' the inputs are shifted to right. This shift register shift one bit per cycle.

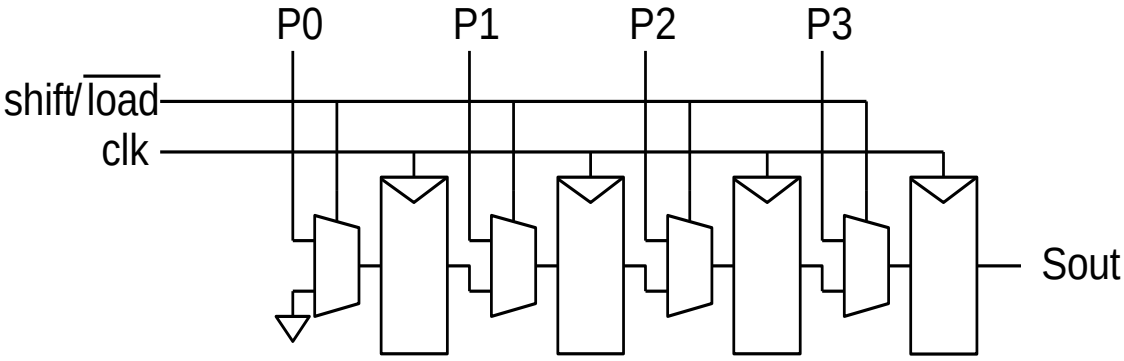

Figure 1.5 Parallel-In-Serial-Out Shift Register

1.2.3.2 Queues

A queue is a pile in which items are added a one end and removed from the other. In this respect, a queue is like the line of customers waiting to be served by a bank teller. As customers arrive, they join the end of the queue while the teller serves the customer at the head of the queue. The major advantage of queue is that they allow data to be written at different rates. The read and write use their own clock and data. There is an indication in queue when it is full or empty. These kind of queues usually built with SRAM and counters. There are two types of queues they are First-In-First-Out and Last-In First-Out.

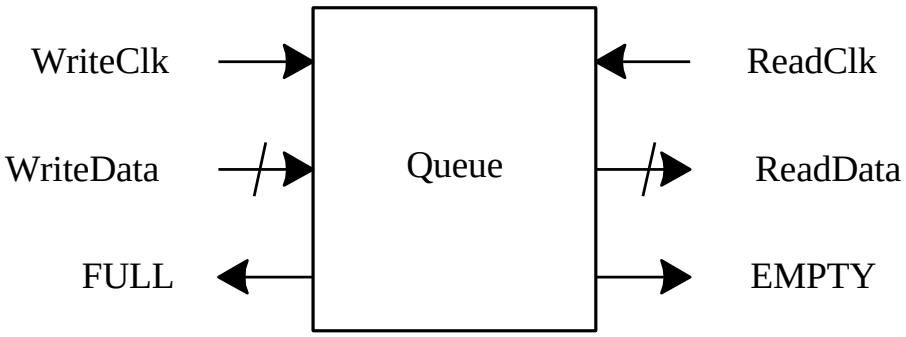

Figure 1.6 Queue

First-In-First-Out: In this method initialize the read and write pointers to the first element. Check whether the queue is empty. In write mode we will assign the write pointer and increment the write pointer. If the write almost catches read then queue is full. In read mode we will increment the read pointer.

Last-In-First-Out: It is also called as stack; objects which are stored in a stack are kept in a pile. The last item put into the stack is at the top. When an item is pushed into a stack, it is placed at the top of the pile. When an item popped, it is always the top item

which is removed. Since it is always the last item to be put into the stack that is the first item to be removed, it is last-in, first-out.

1.2.4 Contents-Addressable Memory (CAM)

It is another important classification of nonrandom access memories. Instead of using an address to locate a data CAM [16] uses a word of data itself as input in a query-style format. When the input data matches a data word stored in the memory array, a MATCH flag is raised. The MATCH signal remains low if no data stored in the memory corresponds to the input word. This type of memory is also called as associative memory and they are an important component of the cache architecture of many microprocessors.

The Figure 1.7 is an example of 512-word CAM architecture [16]. It supports three modes of operation read, write and match. The read and write modes access and manipulate the data same as in an ordinary memory. The match mode is a special function of associative memory. The data patterns are stored in the comparand block which are needed to match and the mask word indicated which bits are significant. Every row that matches the pattern is passed to the validity block.

The valid rows that match are passed to the priority encoder leaving the rows that contain invalid data. In the event that two or more rows match the pattern, the address of the row in the CAM is used to break the tie. In order to do that priority encoder considers all the 512 match lines from the CAM array, selects the one with the highest address, and encodes it in binary. Since there are 512 rows in CAM array, it needs 9 bits to indicate the row that matched. There is a possibility that none of the rows matches the pattern so there is one additional ‘match found’ bit provided.

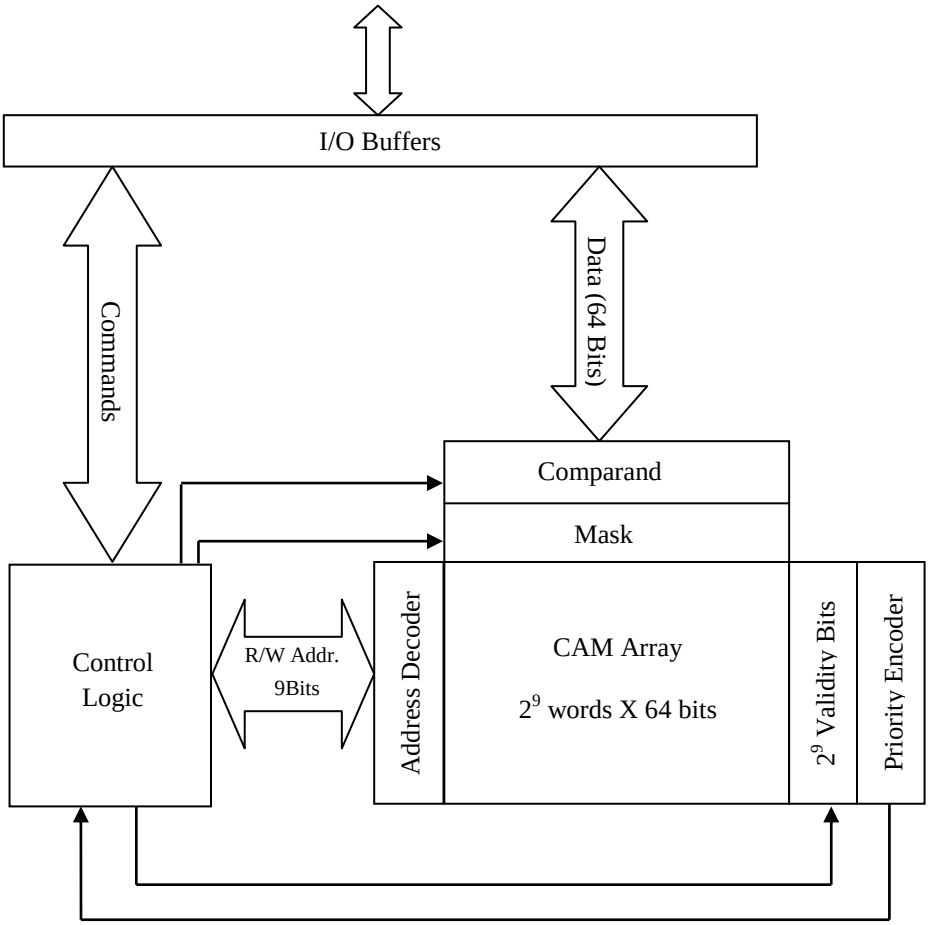

Figure 1.7 Architecture of 512-word CAM [Ref .16]

1.3 SRAM Architecture

Memory organization is very important in every application it can be either portable wireless devices or system on a chip. Generally memories consume most of the power. So it comes immediately that memories have to be designed hierarchically. It is not possible to utilize for a memory the maximum speed and capacity at lowest cost and power [9]. For immediate utilization, data can be stored in expensive registers like cache memories and less used data in large memories like hard disk.

The choice of memory architecture is very important for each application. For a best tradeoff between power, delay and maximum utilization of memory on has to think to hierarchical, parallel, interleaved and cache memories [9]. The architecture of the SRAM is based on the precharge of the bitlines. The precharge and access time can be very long for larger memories without any speed-up techniques.

The typical SRAM design is shown in figure 1.8 the memory array contains the memory cells which are readable and writable [10]. The Row decoder selects from 1 out of n = 2k rows, while the column decoder selects $l = 2^i$ out of $m = 2^j$ columns. The addresses are not multiplexed as it in the DRAM. Sense amplifier detects small voltage variations on the memory complimentary bitline which reduces the reading time. The conditioning circuit is used to pre-charge the bitlines.

The access time is determined by the critical path from address input to the data output as shown in figure 1.9 [10]. This path contains address input buffer, row decoder, memory cell array, sense amplifier and out buffer circuit. The critical delay components are word-line decoding and bitlines sensing delay time. The swing on the bitlines should be as small as possible in order to reduce the sensing time during a read operation.

In a read operation, the bitlines are precharged to some reference voltage usually close to the supply voltage. When word line turns high, the access transistor connected to the node storing ‘0’ starts discharging the bitline while the complementary bitline remains in its precharged state, resulting in a differential voltage between the bitline pair. Since the SRAM has an optimized area results in a small cell current and slow bitline

discharge rate. In order to speed up the RAM access, sense amplifiers are used which amplify the small bitline signal and eventually drive it to the external world.

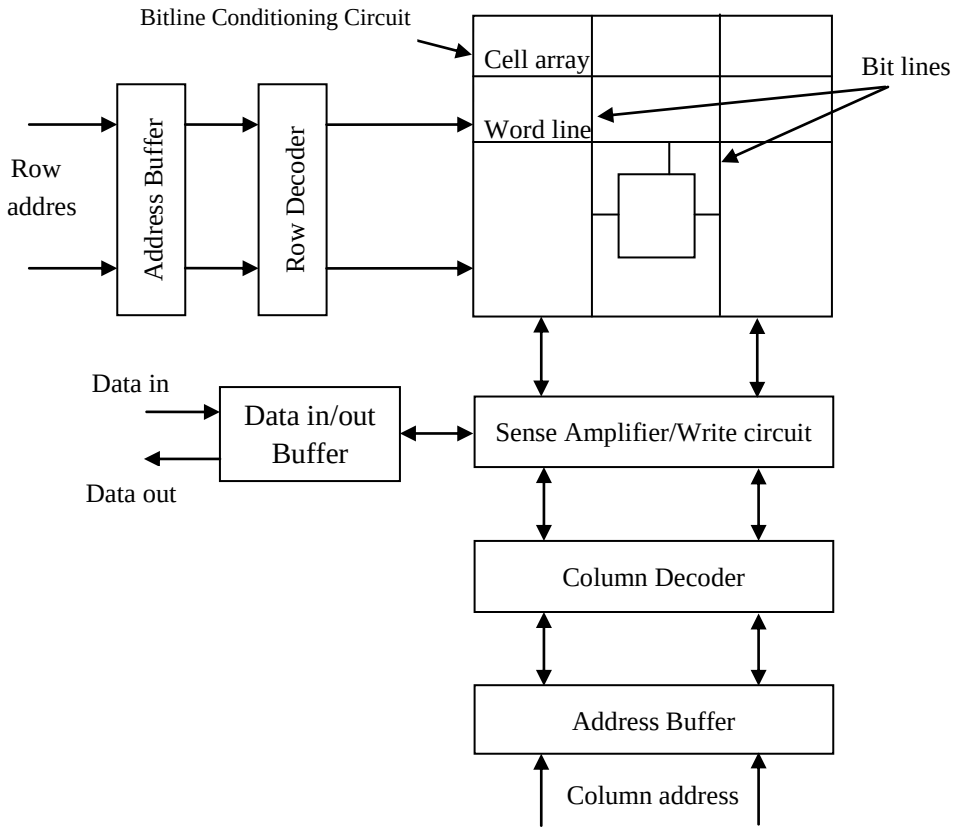

Figure 1.8 Typical SRAM Architecture [Ref. 10]

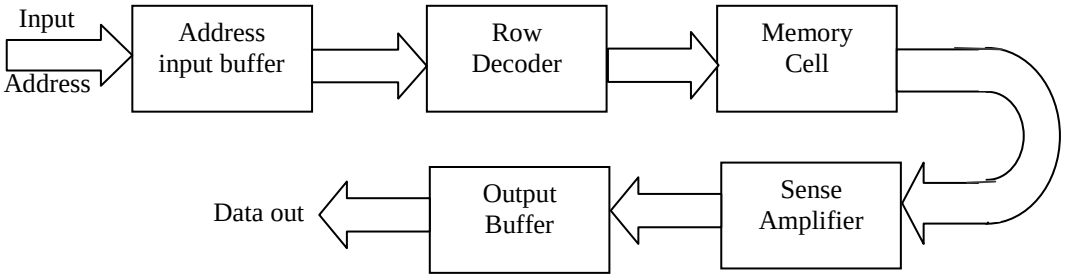

Figure 1.9 Critical path for read access in SRAM

In write operation, the write data is transferred to the desired column by driving the data onto the bitline pairs and selecting a particular row by using word line. If the cell data is different from the write data, then the '1' node is discharged when the access transistor connects it to its discharged bitline, thus causing the cell to be written with the bitline value.

The major two portions which explain about the purpose of design and optimization for access path are the row decoder – the path from the address inputs to the word line, and the data path - the portion from the memory cell portions to the SRAM I/O ports. While designing row decoder two major aspects to have to be considered, firstly determining the optimal circuit style and the optimal sizing of the circuits and secondly amount of buffering at each level. Basically NOR gates are used to design the decoders, which we have discussed briefly in chapter 2.

Chapter 2 explains about design of 6T SRAM, its sizing and also design of 8-Bit SRAM using 6T cell and 7T SRAM cell. The design of row decoder, write switch and sense amplifier are critical. The sizing's of these devices are to be done carefully for a proper output. In chapter 2 we have also explained about design of 7T SRAM. The row decoder design for 8-Bit 7T SRAM is different from what we have used in 8-Bit 6T SRAM cell as we have two additional wordlines to be considered. The design of 8-Bit 7T SRAM cell and its output waveforms are also discussed in chapter 2.

The static noise margin of 6T and 7T SRAM cell are discussed and compared in chapter 3. The delay comparison is made between 8-Bit 6T and 8-Bit 7T SRAM cell. Chapter 3 also explains about statistical data analysis, process and mismatch variations.

Statistical data analysis is recorded for 8-Bit 6T and compared with 8-Bit 7T SRAM cell. Finally we have summarized the conclusion of this thesis at the end of chapter 3

Design of 8-bit SRAM using 6T and 7T cells

2.1 Six Transistor Memory cell

The memory cell plays an important role in design of low power and high density SRAMs because the memory size is dominated by the cell area. There are various static memory cells. The basic six transistor memory cell is shown in Fig 2.1[10] which is in the form of two inverters, cross coupled with two pass transistors connected to complementary bitlines *BL* and $\overline{BL}$. The pass-transistors are controlled by the signal WL (word-line).

During the Read cycle, the bit-lines *BL* and $\overline{BL}$ are held high (precharged). Assume that '0' is stored at node A and '1' is stored at node B. when the cell is selected i.e., *WL* set to '1' either *BL* or $\overline{BL}$ will be discharged.

In the write mode, one of the bitlines is pulled low and the other high and then the cell is selected by *WL*. Assume that $\overline{BL}$ is set to '0' while initially '1' is stored at node A. *N1* and *P1* are sized such that node A is pulled down enough to turn *P2* on. This in turn causes node B to be pulled up. The cross coupled inverter pair has a high gain to cause nodes A and B to switch to opposite voltages. The data retention current (standby) of this cell can be as low as 10^{-15} *A.*

The stability of the memory cell is its ability to hold the state. Static noise margin is the DC disturbance, such as offsets and mismatches due to processing and variations in process conditions. The SNM is defined as the maximum value of V_n that

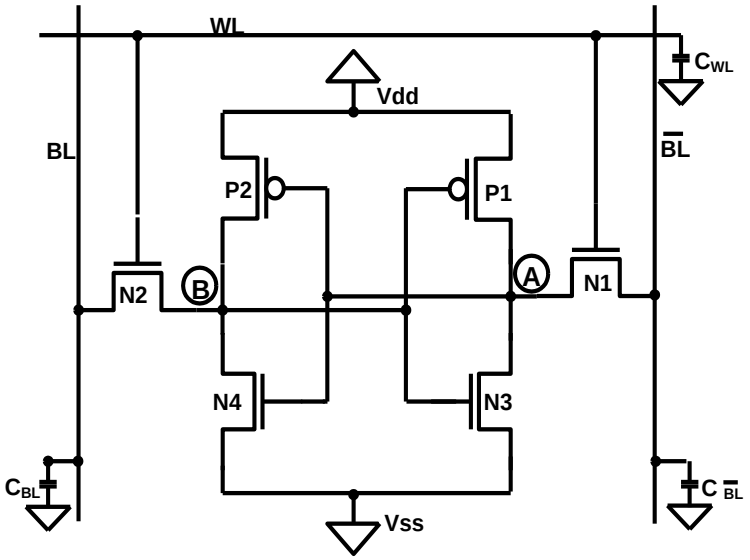

Fig 2.1 Six Transistor Memory Cell

can be tolerated by the cross coupled inverter before altering state. An important parameter in SNM is memory cell ratio, r, defined by

$$r = \frac{(W_{n4}/L_{n4})}{(W_{n2}/L_{n2})}$$

Where transistors *N2* and *N4* are the access and driver nMOS transistors shown in Figure 2.1. This Static noise margin increases with the ratio r. However it is limited by the cell area constraint. The stability of the cell is maintained even if the V_{DD} is scaled down.

Six-transistor SRAM, cell needs a careful device sizing for read stability, write-margin and data retention. Pull-down nMOS is made stronger than pass-transistor to achieve good read stability by minimizing the read disturbance. A strong pull-up pMOS improves the read stability, but it also degrades the write margin. Therefore the sizing of the pass-transistor is needed to achieve a good write-margin.

2.2 Seven Transistor Memory cell

The seven transistors memory cell is designed by adding a data protection nMOS transistor N5 between node B and transistor N2 [1]. While SRAM cell is being accessed, $\overline{WL}$ is in activated state, '0' and N5 is OFF. Since N5 prevents the voltage at Node B from decreasing, the data bit is not reserved even if Node A voltage greatly exceeds shown in Figure 2.3.

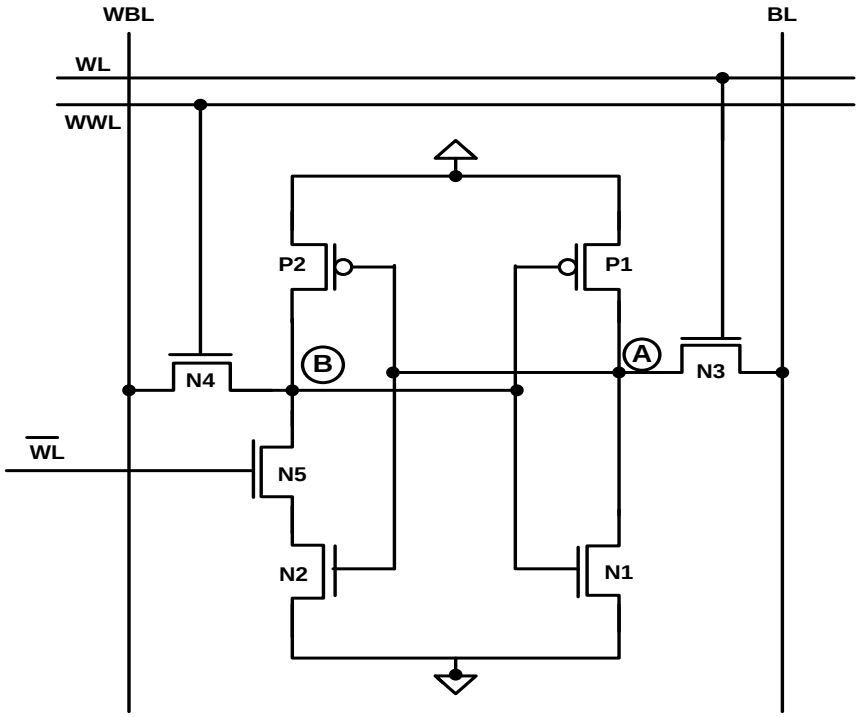

Figure 2.2 Seven Transistor Memory Cell

Seven transistor SRAM consists of three wordlines which are WWL, /WL, WL and two complementary bitlines BL and WBL. /WL and WL are complementary and /WL is used to control the wordlines WWL and WL. The write operation for SRAM 7T is same as that of 6T SRAM cell. In write operation one of the bitline is charged and the other

discharged and the wordlines WWL and WL are closed in order to write in the nodes A and B.

During data retention period, when the SRAM cell is not being accessed, word line signal $\overline{WL}$ is '1' and nMOS transistor N5 is ON. The use of two CMOS inverters results in high cell stability. During read operation, the logical threshold voltage of the CMOS inverter driving node B increases greatly when the data protection nMOS transistor N5 is turned off. For this reason, the read value at A = '0' remains large even when access nMOS transistor N3 is turned on and node A voltage increases.

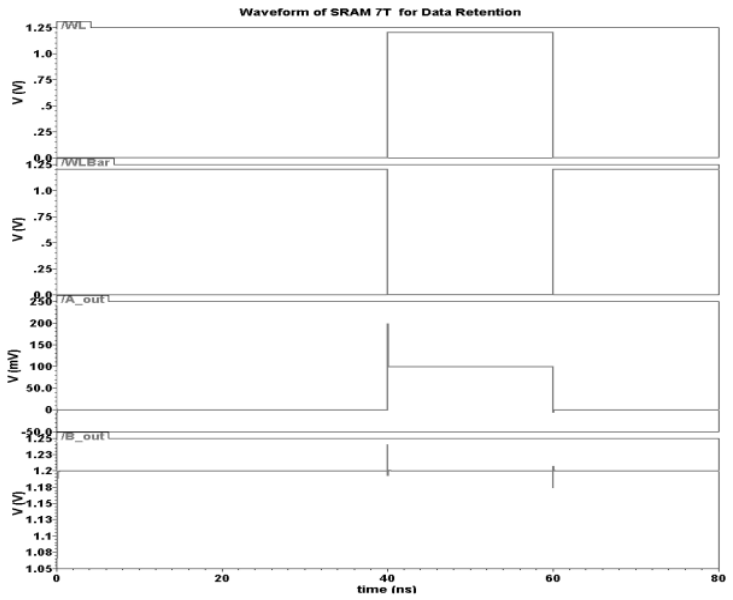

Figure 2.3 Waveform of 7T SRAM cell showing Data Protection by N5 Transistor

The voltage dividing effect takes place at the inverter which stores '0', will be pulled up. In order to stop this transition the 7th transistor at the other node is turned off so that the node which stores '1' will not be pulled down by the driver transistor as it acts a switch between the node and the driver transistor. In read operation the SRAM cell is isolated from the bitlines and the bitlines are precharged to VDD. When the wordlines are

closed the bitlines which is connected to the node which stores ‘0’ will be discharged through pass transistor while the other bitline stays high.

2.3 8-BIT SRAM Architecture

The 8-bit SRAM architecture is described in Figure 2.4 [11]. The 8-bit SRAM is designed in a 4 X 2 array as shown in Figure 2.4. Since the array has four rows and two columns, two address bits A_0 and A_1 are needed to define a row. One bit, A_2 is needed to define a column. The output of each row decoder controls a WL, and the output of each column decoder controls a pair of BLs. The decoders are basically two input NORs only the WL whose decoder input equals 00 goes high, turning on the pass devices of all cells on the WL. All other WLs remain low, isolating their cells from the bitlines. The Row decoder also has a 2 X 4 NAND Decoder, A_0 and A_1 switches. The pMOS transistors are used as switches between V_H and the memory cell. The input pins S_0 & S_1 are used to control the 2X4 decoder which in turn controls the pMOS switches. This function accomplishes the row selection.

Row address decode allows all cells in the row to access their BLs. The column address decodes then specifies the pair of BLs to be connected to the I/O buses by turning on their bit switches through column decoders. Once selected, the cell can be either read or written from I/O pins via the I/O buses.

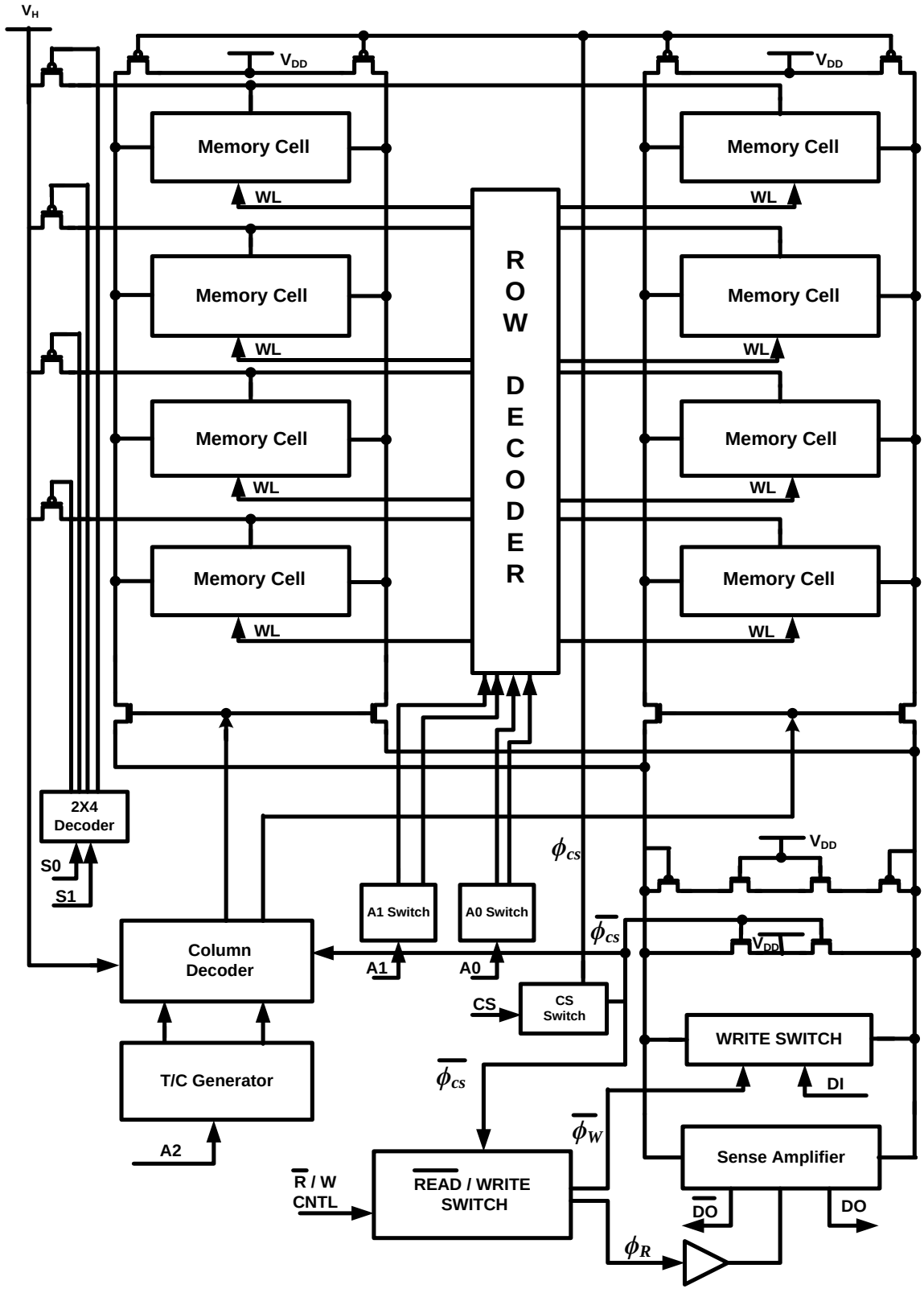

Figure 2.4 SRAM Architecture [Ref.11]

The write switch and the sense amplifier are connected to the I/O buses. During a write operation (WRITE), the write switch will charge one bitline and discharge the other in accordance with the input data: *DI*. Since the write switch does not dissipate DC power and there is only one circuit per DI, switch devices can be designed with a large β to ensure fast performance. Indeed, chip performance is seldomly limited by WRITE.

In read operation (READ), the write switch is disabled and the bitlines and the I/O buses are charged/discharged by the memory cell. The rise/fall of the I/O buses will then be detected by sense amplifier. If the number of cells connected to each bitline and the number of bitline pairs multiplexed to the I/O buses are both small enough that the parasitic capacitances from the cell to the sense amplifier are small, the cell will force the sense amplifier to conform to its state.

For SRAMs of practical sizes, however, the loading on the bitlines and I/O buses are much too heavy to be affected by a small cell; READ can be very slow. Furthermore, if the residual charges left on the bitlines by the previous operations are of the opposite polarity, the cell may be overridden by bitlines, resulting in false data. Cell stability then is also a potential problem.

To improve the performance and ensure cell stability, most SRAMs precharge and equalize the bitlines and I/O buses before the cell is accessed. Performance is improved by the fast discharge through the pulldown devices.

During precharge phase the bitlines and I/O buses are precharged to high, equalizing the bitlines to V_{DD}. A gating pulse ϕ_{cs} is needed to enable the precharge phase. As we can see from the Figure 2.4, chip select is inverted and applied to the precharge devices

placed above write switch. Prior to the chip select,$\overline{\phi_{cs}}$ is high, turning on precharge devices and precharging the bitlines and I/O buses to (V_H - V_T). As soon as ϕ_{cs} goes up the precharge devices turn off, switching control of bitlines to the cells.

In the RAM Specifications, the time elapsed from chip to output data valid is called *chip select access time*, t_{ACS}, and the minimum time between any two consecutive operations is called *chip cycle time* t_{CL}. Since the chip has to be "deselected" in each cycle to allow bitline restore, the chip cycle time is the sum of the access time and the restore time.

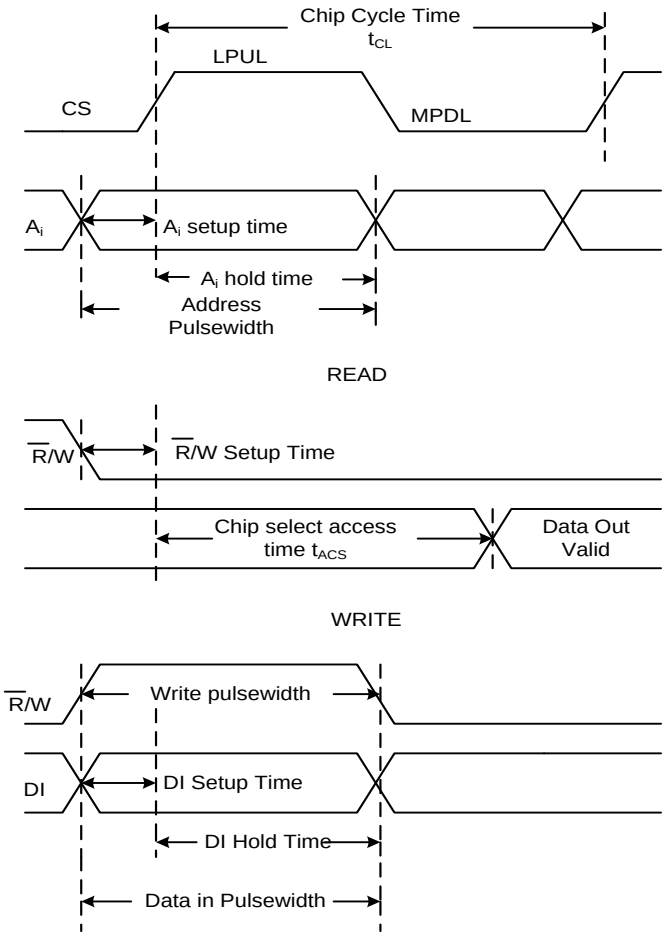

Figure 2.5 Chip I/O signal timing

2.4 Circuit Implementations

The circuit shown in Figure 2.4 is the simplest and perhaps the fastest implementation of SRAMs. Most circuits are static. Static circuits do not make use of temporary charges on a capacitor to maintain voltage levels. Since there are no precharge/discharge requirements, static circuits can operate without clocks. All voltage levels are held indefinitely as long as the inputs do not change. The only timing limitation is the normal circuit delay.

In this section we shall discuss the various functional blocks which are shown in the above figure.

2.4.1 True/Compliment Generator: *T/C GEN:*

Both CS (chip select) and A_i (Inputs A_0, A_1, A_2) are needed in true/compliment form. The first circuits they encounter are therefore the T/C GENs. The T/C GEN is designed by using inverters as shown in figure. Simple inverter implementation minimizes circuit delay and hence the access time of the chip. Due to lack of memory, however, it also imposes a minimum pulsewidth requirement on its input. For example, in a READ the Ai's must be held constant until the sense amplifier latches up.

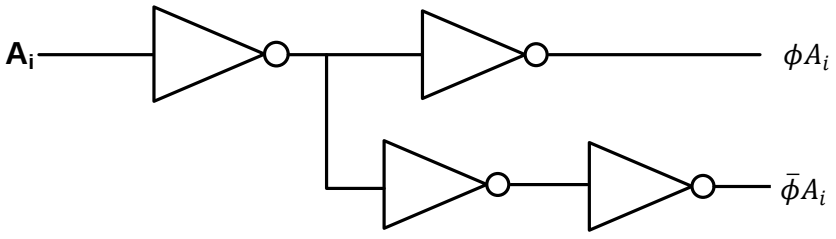

Figure 2.6 T/C Generator

Since one of the inverters is always on (output low), the DC power dissipation is high. Various dynamic circuits can be used to cut down the DC power in many practical designs. At the same time, however, dynamic circuits require more elaborate timing and usually run slower than their static counterparts. The T/C GEN circuit used here is fully static and does not dissipate DC power.

2.4.2 Decoders (*DEC):*

From a logic point of view, a decoder can be implemented either with multiple NORs or Multiple ANDs. Here in we have designed using multiple NOR gates. The switches A_0 and A_1 are used in order to select the row and A_2 is used to select the column. The output of the decoder is connected to the wordlines, as they are in turn connected to the gates of the pass devices.

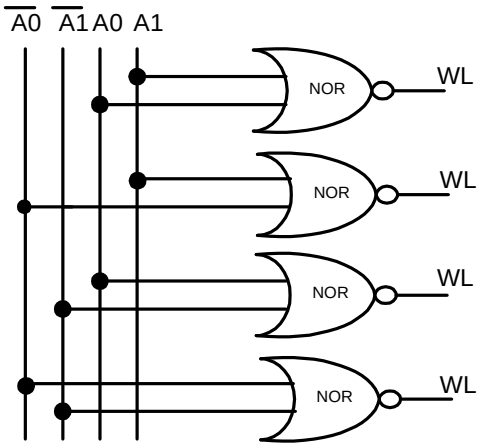

Figure 2.7 Row Decoder for 6T 8-Bit SRAM

The inputs S_0 and S_1 are used to control the decoder, the outputs of the decoder are connected to the gates of the pMOS transistors. These pMOS transistors act as switches between the supply voltage V_H and the memory cells. The decoder which we use here is

made of NAND gates, as when the inputs are high the output will be low and hence the pMOS conducts. This method can be implemented when we consider for low power circuit design, as we can see that at a time there will be voltage supply for only one row, and that is the row where we need to perform the READ or WRITE.

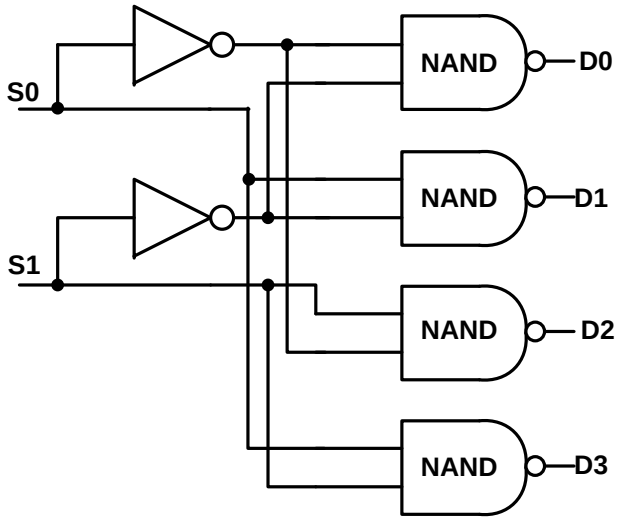

Figure 2.8 NAND Decoder

The reason for using NAND circuit is because n-channel devices are considerably faster than p-channel devices, NAND circuit that stack n-channel devices are faster than NOR circuit which stack p-channel devices.

The row decoder used in 8-bit 7T SRAM cell is different as it has to control two additional wordlines from 8-BIT 6T SRAM cell. The functionality of 8-Bit SRAM and the 7T SRAM cell is explained above. The Figure 2.9 shows the row decoder used for 8-Bit 7T SRAM cell. The $\overline{WL}$ is complement of *WL*. The inputs A_0, A_1 and $\overline{A_0}$, $\overline{A_1}$ are used as inputs to row decoder. The True/complement generator is used to generate the complement of A_0 and A_1. When the input equals to 00 and $\overline{WL}$ equals '0' then WL and

WWL goes high connecting the bitlines to the SRAM cell as the wordlines WL and WWL are connected to the pass transistors of SRAM cell. As this row decoder is designed with NOR gates the wordlines whose decoder input equals 00 goes high and all other wordlines remains low isolating their cells from the bitlines.

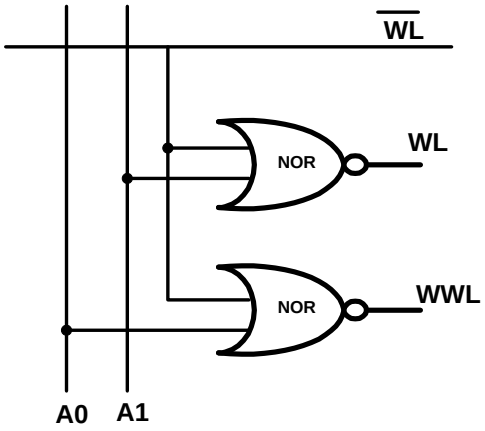

Figure 2.9 Row Decoder for 7T SRAM cell.

2.4.3 Read/Write Control: $\bar{R}/W$ CNTL:

The $\bar{R}/W$ CNTL decodes the $\bar{R}/W$ command. With ϕ_{cs} high, $\bar{R}/W$ CNTL either activates the sense amplifier by pulling up ϕ_R or releases the write switch by pulling down $\bar{\phi}_w$. The logic equation is given by

$$\phi_R = \phi_{cs}\,\bar{R}/W \qquad \& \qquad \phi_w = \phi_{cs}\bar{\phi}_R = \overline{\bar{\phi}_{cs} + \phi_R}$$

Since $\bar{R}/W$= 0 corresponds to READ and $\bar{R}/W$ = 1 corresponds to WRITE, the input command is designated as $\bar{R}/W$, which is referred to as "negative READ/positive WRITE."

Since it is not latched, the $\bar{R}/W$ pulse must be held constant until the completion of the READ/WRITE operation. Furthermore, both *DI* and $\bar{R}/W$ must precede ϕ_{cs} because any changes of *DI* or $\bar{R}/W$ after ϕ_{cs} has already gone up will invalidate the access and cycle time specifications.

2.4.4 Sense Amplifier:

The sense amplifier (SA) is an important circuit to regenerate the bit-line signals in a memory design. The SA can be also applied to the receiving of long interconnection signal with large RC delay and large capacitive load signal. Moreover, the complexity of the differential logic circuit can be enhanced by combining the SA with differential logic networks to reduce the delay time.

Figure 2.10 shows the circuit diagram of one-column SRAM. The SRAM cell provides a very small read current for small area consideration. As the capacity of the memory increases, the number of SRAM cells on one column increases. Therefore, an efficient SA design is necessary to improve the speed limitation for the decreasing of bit-line load.

In READing, signals on the bitlines are latched up by the sense amplifier which is shown in Figure 2.11. After the address decoding is completed, one of the bit switch pairs open and differential voltage starts to develop across the I/O buses. The set pulse $\phi_R(t - \tau)$ then goes up, turning on the devices which are connected to pulldown the *DO* and $\overline{DO}$. With the differential voltage in the I/O buses, the internal regenerative action of the latch amplifies the voltage difference and drives the latch to one of its equilibrium state. Ideally the set pulse $\phi_R(t - \tau)$ should come up right after, but not prior to, the pass device

opening of the cell selected. The delay time τ from $\phi_R\,(t)$ has to be carefully controlled. On the one hand, τ must be long enough to avoid premature latch setting. On the other hand, an overly long delay affects the access time.

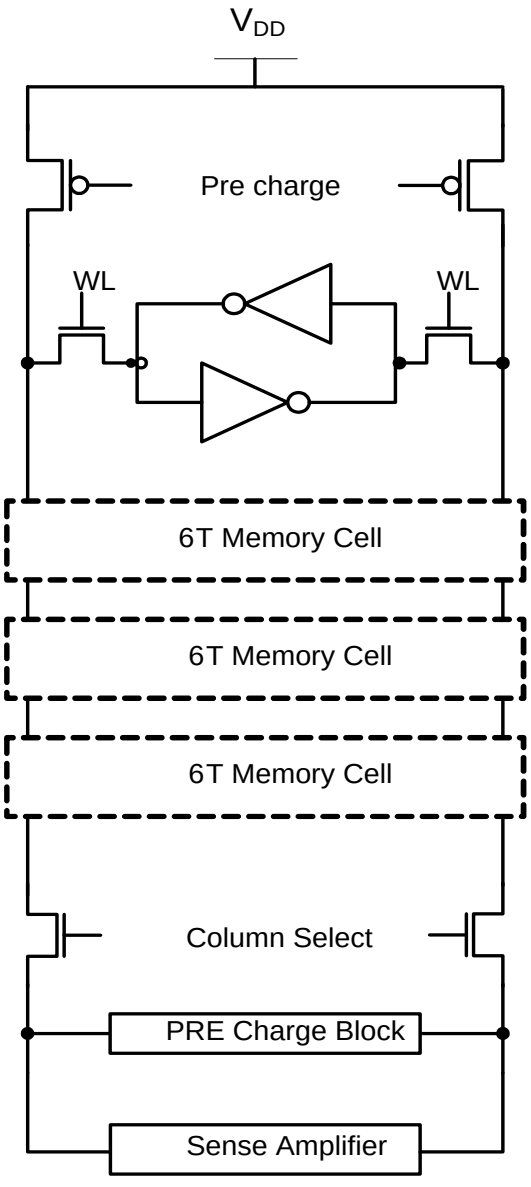

Figure 2.10 Sense Amplifier

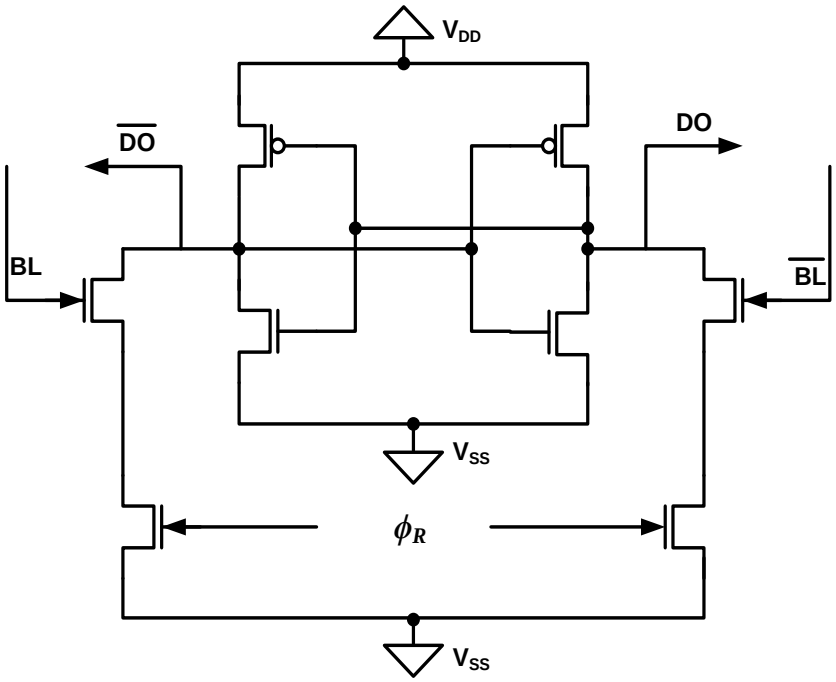

Figure 2.11 Differential sense amplifier

The design of the sense amplifier depends on the strength of the signal. If the signals are fast and strong, the simple latch can latch up with little delay. However, if we have a design which has many cells in one particular column then we need to go for more sophisticated sense amplifier.

2.4.5 Write Switch:

In the WRITE mode $\overline{\phi_W}$ is low, releasing the write switch to the input data *DI* and $\overline{DI}$. The transistor sizing of the WRITE switch devices are large to ensure performance.

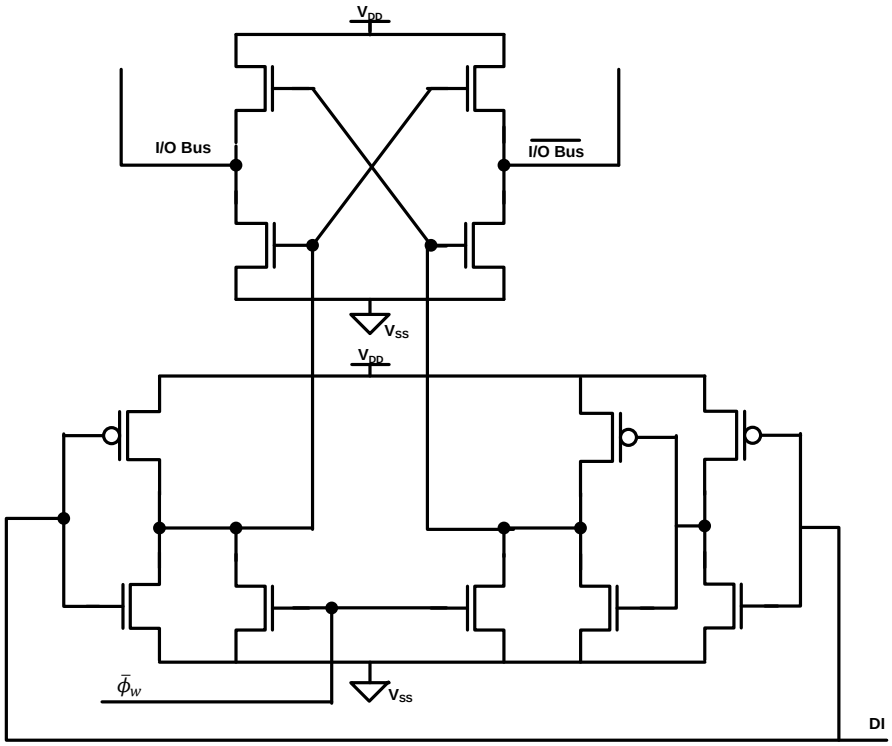

Figure 2.12 Write switch

2.4.6 Holdup Devices:

The control of I/O buses switches from precharge devices to the selected cells during a READ. As $\overline{\phi_{CS}}$ goes down, the turning off of the precharge devices which are connected to the supply voltage charges out of the I/O buses, resulting in a voltage droop. After that if the address decoding cannot completed in time, the I/O buses will be left floating. The potentials on the buses are subject to noise and will continue to fall due to leakage currents.

To stabilize the buses, a pair of pMOS devices is connected in diode configuration and pair of nMOS devices are connected for partial isolation are used to hold the I/O

buses up. These holdup devices, also known as “bleeding devices,” are small enough that they do not interfere with normal operations.

2.4.7 Bitline and I/O Bus Restore:

As soon as the chip is deselected with CS going down, all bitlines and the I/O buses are charged up to $(V_H - V_T)$. Since the load is heavy, it takes some time for $\overline{\phi_{CS}}$ to complete the operation. If $\phi_R\ (t - \tau)$ falls before the I/O buses are equalized, the output of the sense amplifier will stay constant regardless of the restore operation; otherwise, the output is indeterminate.

2.5 Simulation Waveforms for 8-BIT 6T SRAM cell

The output waveforms of the 8-bit six transistors SRAM are shown below. As we can see that in Figure 2.13 the Data Input *DI* is pulse and writes in the memory cell till the $\overline{Read}/Write$ Switch is in write mode. S_0 and S_1 are the select lines and A_0, A_1 are the inputs used to select the row in which the data has to be written and A_2 is the select line which is used to select column.

CS is the Chip select switch, when it is low it turns on the precharge transistors Figure 2.4 in order to precharge the bitlines. *WL* is the Word line which is used to control the access transistors in the memory cell. *DO* and $\overline{DO}$ are the output of the sense amplifier. *BL* and $\overline{BL}$ are the bitlines connected between sense amplifier and the six transistor memory cell.

B_out and *A_out* are the outputs of the memory cell. In the WRITE mode the Data Input is written in the memory cell and in the read mode the data is read through the sense amplifier. The write delay time is calculated through *B_out* and the *DI* and the read delay time is calculated through the *WL* and the *DO* of sense amplifier.

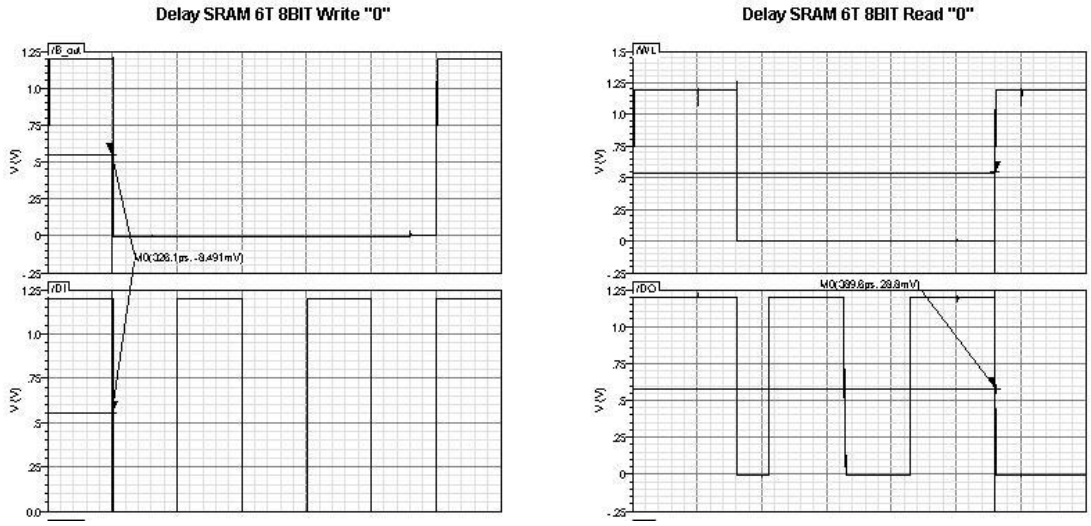

Figure 2.13 Output waveform of 8-bit 6T SRAM with write ‘0’ and read ‘0’ operation.

In Figure 2.13 we can see that Data Input *DI* is written in *B_out* (i.e., node B in Figure 2.1) when $\overline{Read/Write}$ switch is in write mode. The delay time of write operation when performing write ‘0’ in the cell is calculated and the value is 328 ps and the delay of read ‘0’ is 390 ps. The delay time of write ‘1’ operation is calculated between *DI* and *B_out* and the value is 302 ps. The delay time calculated between *WL* and *DO* of sense amplifier is 348 ps.

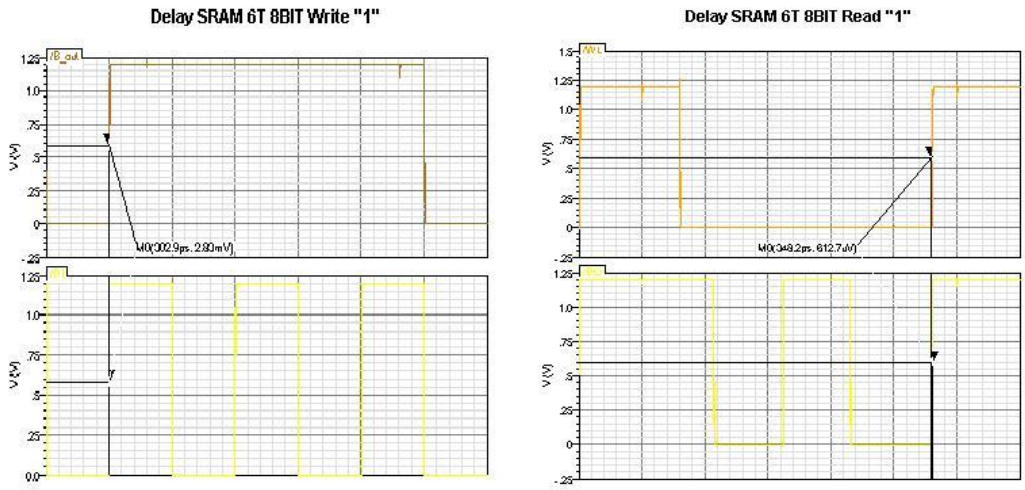

Figure 2.14 Output waveform of 8-Bit 6T SRAM write '1' and read '1' operation

2.6 Simulation waveforms for SRAM 8-BIT 7T cell

The output waveforms of 8-Bit 7T SRAM cell are shown below. The *DI* is the data input of the SRAM, the write switch generates the true/complement of the data input in bitlines. The select lines A_0, A_1 S_0 and S_1 are used to select the row and A_2 is used to select the column. The CS is the chip select signal which controls precharge devices and the $\overline{Read}$/Write Control block. When CS signal goes low the bitlines are precharged to perform read operation.

The *B_out* and *A_out* are the output nodes of 7T SRAM cell. *DO* and $\overline{DO}$ are the outputs of sense amplifier. The write delay is calculated from *DI* to *B_out* to *DO* of SRAM as shown in figure 2.15. The read delay is calculated from wordline WL to the sense amplifier output *DO.* The delay time calculated for write '0' operation as shown from Figure 2.15 is given by 239 ps and the delay time for read '0' operation is given by 320 ps. Figure 2.16 shows the delay time for write '1' operation is given by 265 ps

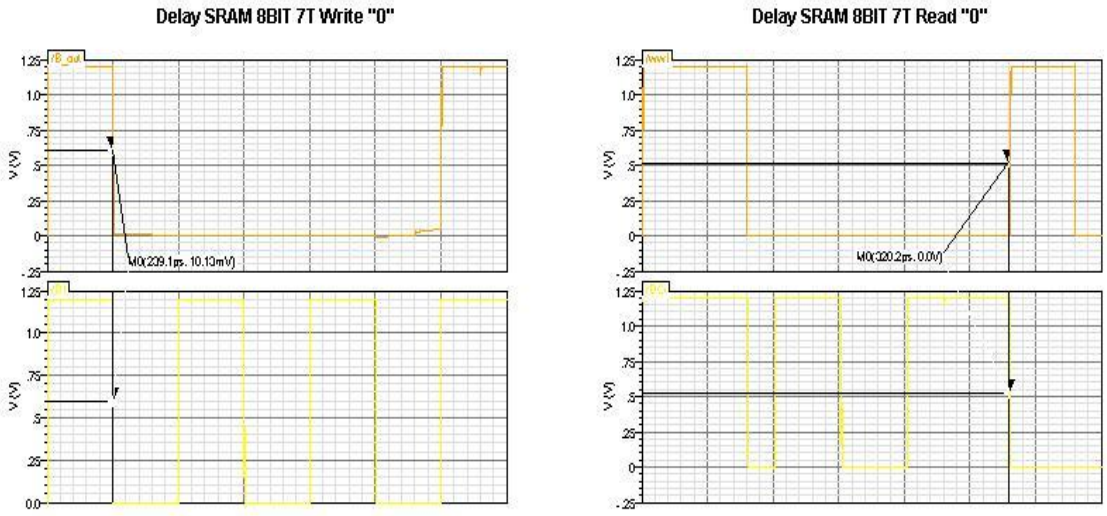

Figure 2.15 Output waveform of 8-Bit 7T SRAM showing write ‘0’ and read ‘0’ operation.

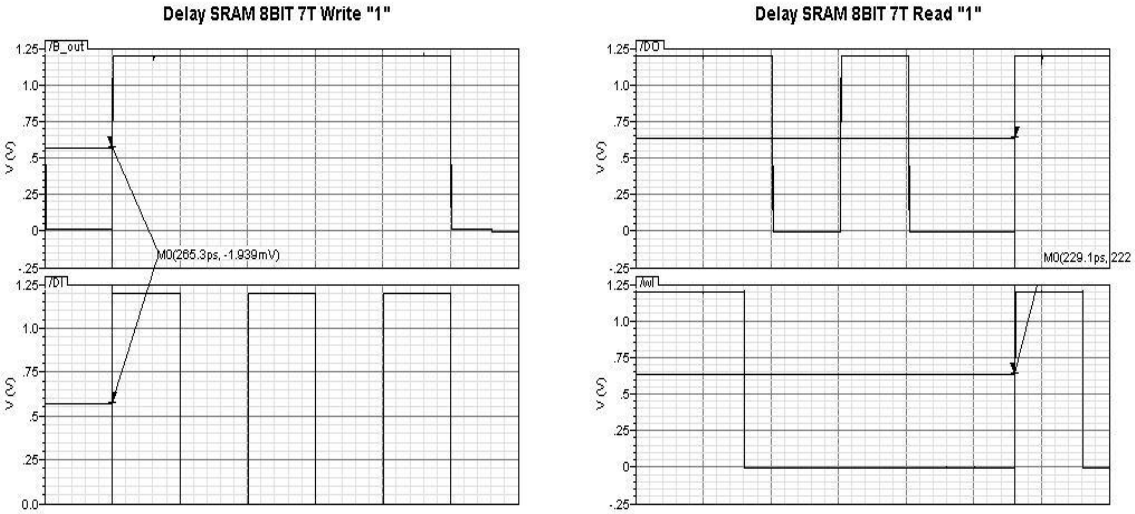

Figure 2.16 Output waveform of 8-Bit 7T SRAM cell showing write ‘1’ and read ‘1’ operation.

and the read ‘1’ operation is given by 229 ps.

Stability and Static Noise Margin Analysis

3.1 Static Noise Margin

The data retention of the SRAM cell in hold state and the read state are important constraints in advanced technologies. The cell becomes less stable at low V_{DD}, increase in leakage currents and increasing variability, "The stability is usually defined by the static noise margin as the maximum value of the DC noise voltage that can be tolerated by the SRAM cell without altering the stored bits."[12]

The read static-noise-margin (SNM) deteriorates with decrease in supply voltage (V_{DD}) [1], [2] and increases with the transistor mismatch. This mismatch happens due to variations in physical quantities of identically designed devices i.e., their threshold voltages, body factor and current factor. Though SNM decreases at low V_{DD} the overall SRAM delay increases and moreover the read operation at low V_{DD} leads to storage data destruction [1] in SRAM cells.

The main operations of the SRAM cell are the write, read and hold. The static noise margin is certainly more important at hold and read operations [13]. Specifically in read operation when the wordline is '1' and the bitlines are precharged to '1'. The internal node of SRAM which stores '0' will be pulled up through the access transistor across the access transistor and the driver transistor. This increase in voltage severely degrades the SNM during read operation.

The SNM deteriorates with the decrease in supply voltage, at low V_{DD} the read and write operations cannot perform properly. The 6T SRAM cell cannot operate efficiently

at low V_{DD} as the V_{DD} decreases the bits which are stored in memory cell can be altered. The SNM of 1-BIT 6T and 7T cells are recorded and compared by varying the V_{DD} and V_{SS} of the cells. The V_{DD} is gradually decreased by 10% of the actual voltage i.e., 1.2V and the V_{SS} is gradually increased by 10%.

In order to overcome the static noise margin at low V_{DD} 7T SRAM cell is implemented. It has one additional transistor compared with 6T cell and operates more efficiently than 6T cell at low-V_{DD}. The 7th transistor which is nMOS transistor is placed in between the node and the driver transistor. The voltage dividing effect takes place at the inverter which stores '0', will be pulled up. In order to stop this transition the 7th transistor at the other node is turned off so that the node which stores '1' will not be pulled down by the driver transistor as it acts a switch between the node and the driver transistor.

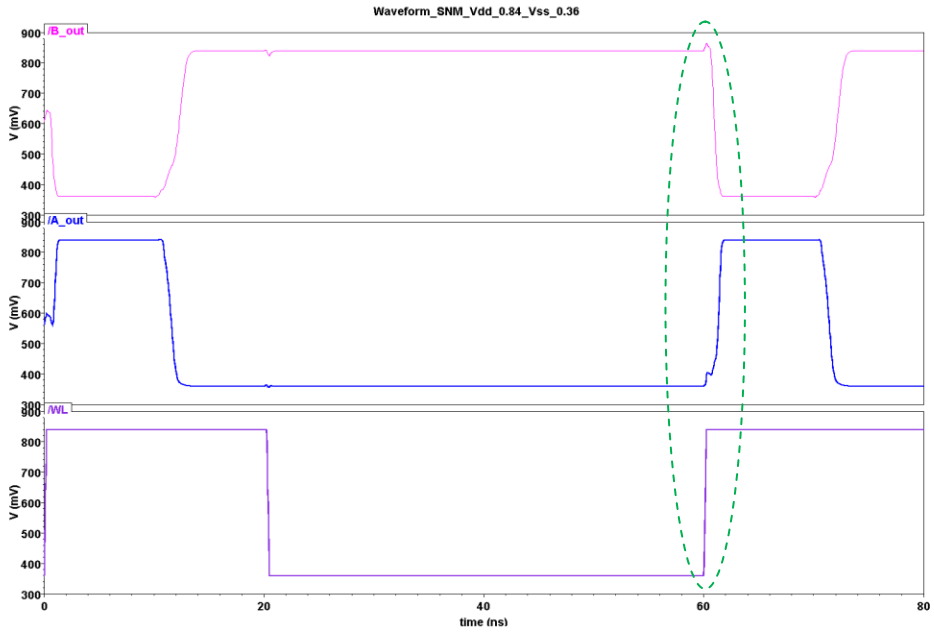

Figure 3.1 Static Noise Margin of SRAM 8BIT 6T at a voltages of V_{DD} = 0.84v and V_{SS} = 0.36v.

The Figure 3.1 shows the stability of 6T SRAM cell at low voltages. In precharge phase when bitlines are charged to V_{DD} and when the wordlines are closed the voltages at node A and B are overwritten by the bitlines. This is called Read-Static-Noise-Margin of SRAM. The Table 3.1 shows the comparison between 6T SRAM and the 7T SRAM at varying voltages.

Table 3.1 Static Noise Margin (SNM) for 1-BIT SRAM Cell

Voltages		**SRAM 6T**	**SRAM 7T**
V_{DD}	**V_{SS}**	**Read Delay (ps)**	**Read Delay (ps)**
1.2	0	10.5	5.6
1.08	0	16.3	10.22
1.08	0.12	25.94	15.36
0.96	0.12	47.65	27.62
0.96	0.24	55.68	33.02
0.84	0.24	64.13	58.53
0.84	0.36	-	68.36
0.72	0.36	-	80.4

Though 7T SRAM can perform better than 6T at low V_{DD} it has limiting factors:

1. The Table 3.1 shows the comparison is made till V_{DD} = 0.72v and V_{SS} = 0.36v these are the maximum low voltages that this 7T SRAM can perform according to the transistor sizing. If we operate 7T SRAM lower than the stipulatedvoltage levels then the write operation cannot be performed, since the write margin decreases with decreasing V_{DD}.

2. Read operation at low-V_{DD} levels result in storage data destruction in SRAM cells this is due to the lealage current of pMOS transistor P2. This is shown in Figure 3.1 as the 6T SRAM cell is operated at low V_{DD}.

3.2 Write and Read operation waveforms

A comparison is made between SRAM 8-BIT 6T and 7T cell write and read operations. As we see that the delay values for 7T SRAM cell is better than that of 6T cell this is due to additional transistor in 7T SRAM. The sizings for all other transistors are similar for both the cells. The write delay is calculated from Data-in DI to the node B of the SRAM and the read delay is calculated from the wordline WL for 6T and WWL for 7T to the sense amplifier output DO.

Table 3.2 Comparison between SRAM 8-BIT 6T and 7T write and read operation

SRAM 8BIT	Write Operation	Delay (ps)	Read operation	Delay (ps)
SRAM 6T	Write '0'	326	Read '0'	389.6
SRAM 7T		239		320.2
SRAM 6T	Write '1'	303	Read '1'	348.2
SRAM 7T		265		229.1

The percentage decrease in delay as we calculated from Table 3.1 as for write '0' operation it is given as 26% and for read '0' operation is 17% and for write '1' operation it is given as 12% and for read '1' operation as 34%. The average percentage decrease for

write operation is calculated as 20% and the average percentage decrease for read operation is calcualted as 25%. The Figure 3.2(a) shows the select lines S_0, S_1 & A_0, A_1,

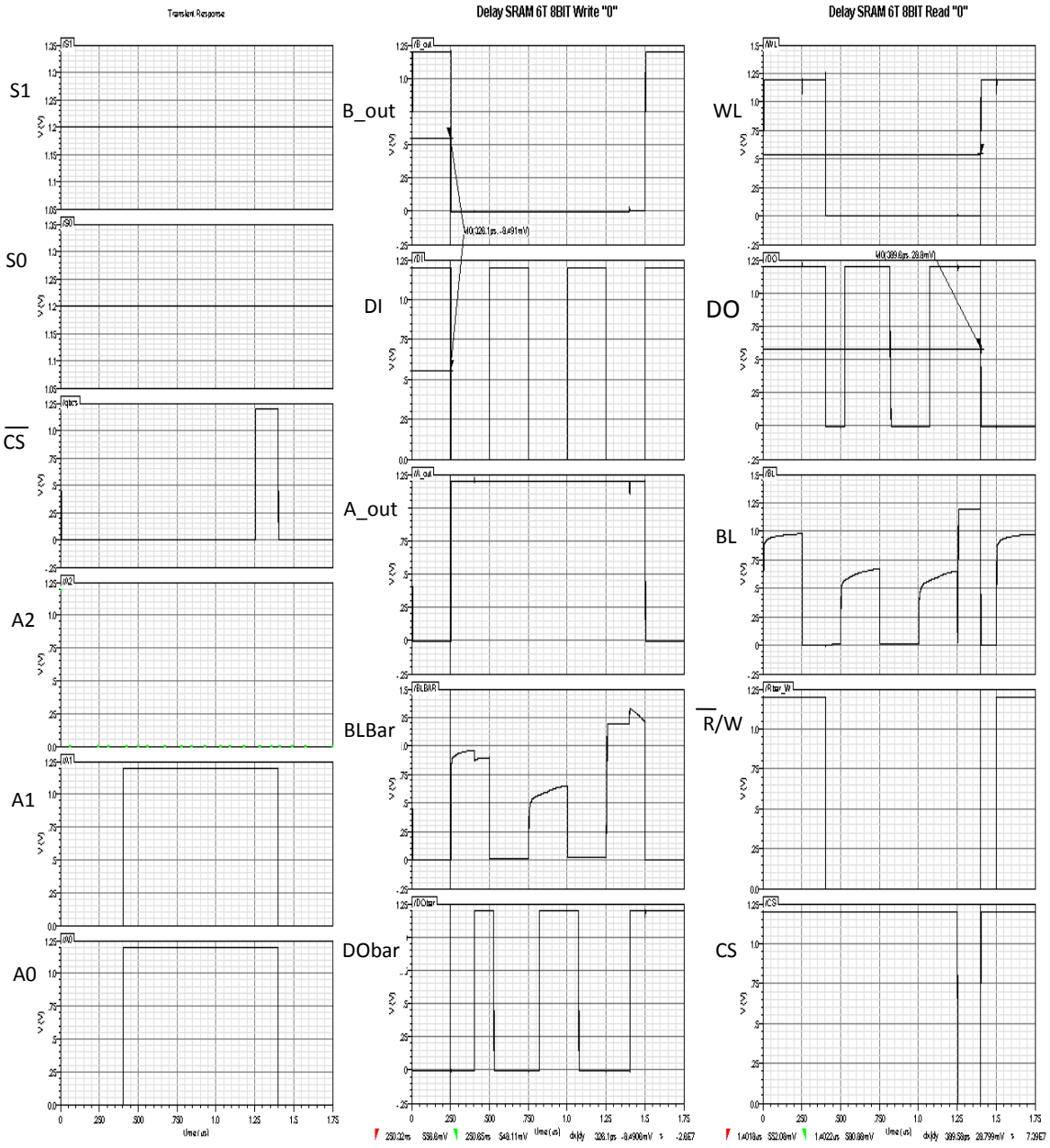

Figure 3.2 Simulation Waveforms for write '0' and read '0' operation of 6T 8-bit SRAM Cell.

A_2, as the inputs to decoder. The signal qbcs is complement of cs (chip select) signal. The figure 3.2(b) shows nodes of SRAM B_out, A_out and DI is the Data input of SRAM. DO (Data output signal) is the output of sense amplifier and DObar is the complement

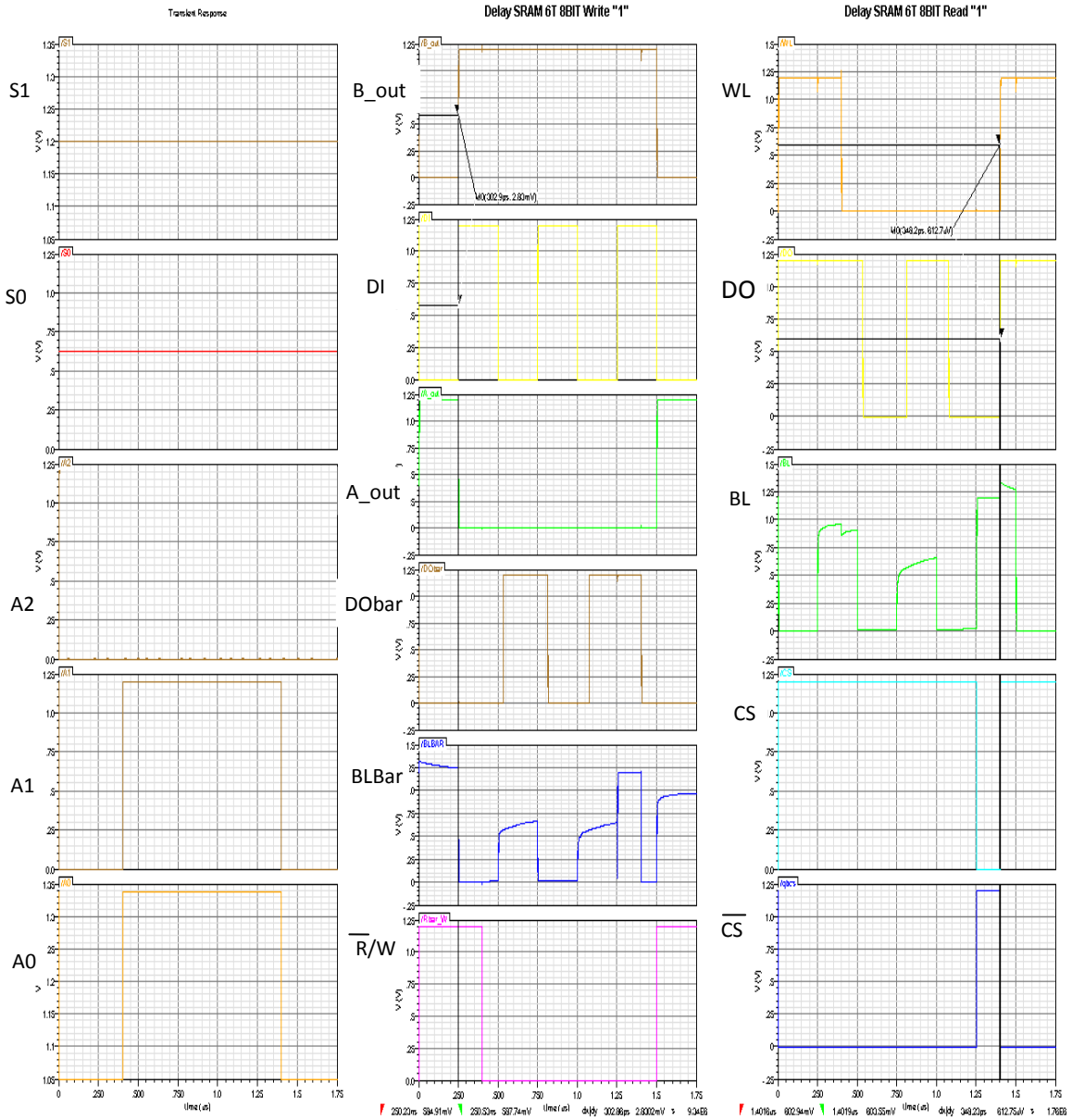

Figure 3.3 waveforms showing write '1' and read '1' operation of 6T 8BIT SRAM cell.

of DO signal. The figure 3.2 (c) shows the signals WL as wordline and BL as bitline and BLbar is complement of BL. Rbar_W is the $\overline{Read}$/Write signal as we have discussed in chapter 2.

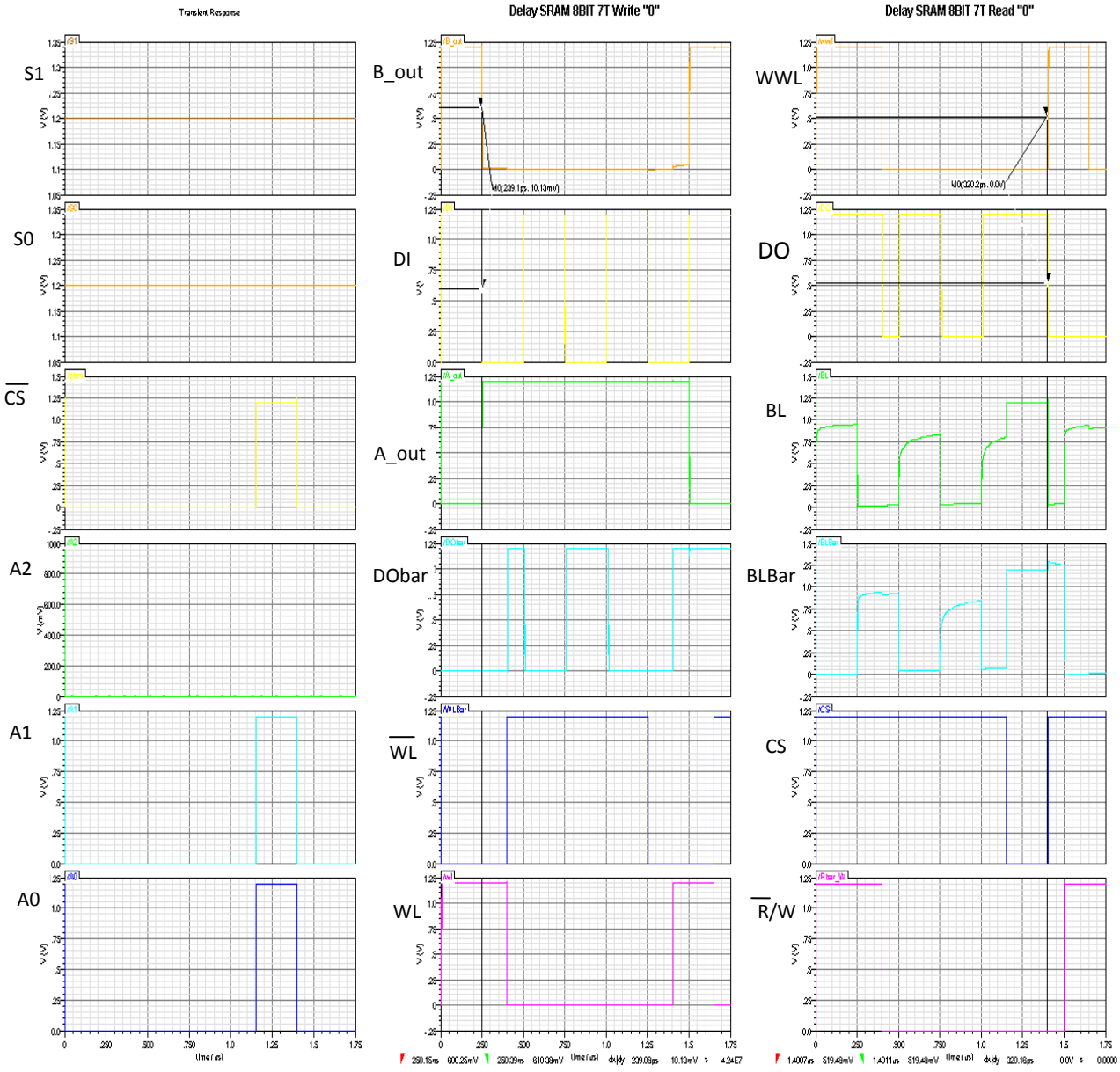

Figure 3.4 Waveforms showing write '0' and read '0' operation of 7T 8BIT SRAM cell.

The explanation for Figure 3.4 is same as that of figure 3.2, the difference is that in figure 3.4 (b) has WLBar which is an additional wordline explained in chapter 2. The Figure 3.4 (c) has WWL which is word-word-line and is used to calculate the read delay.

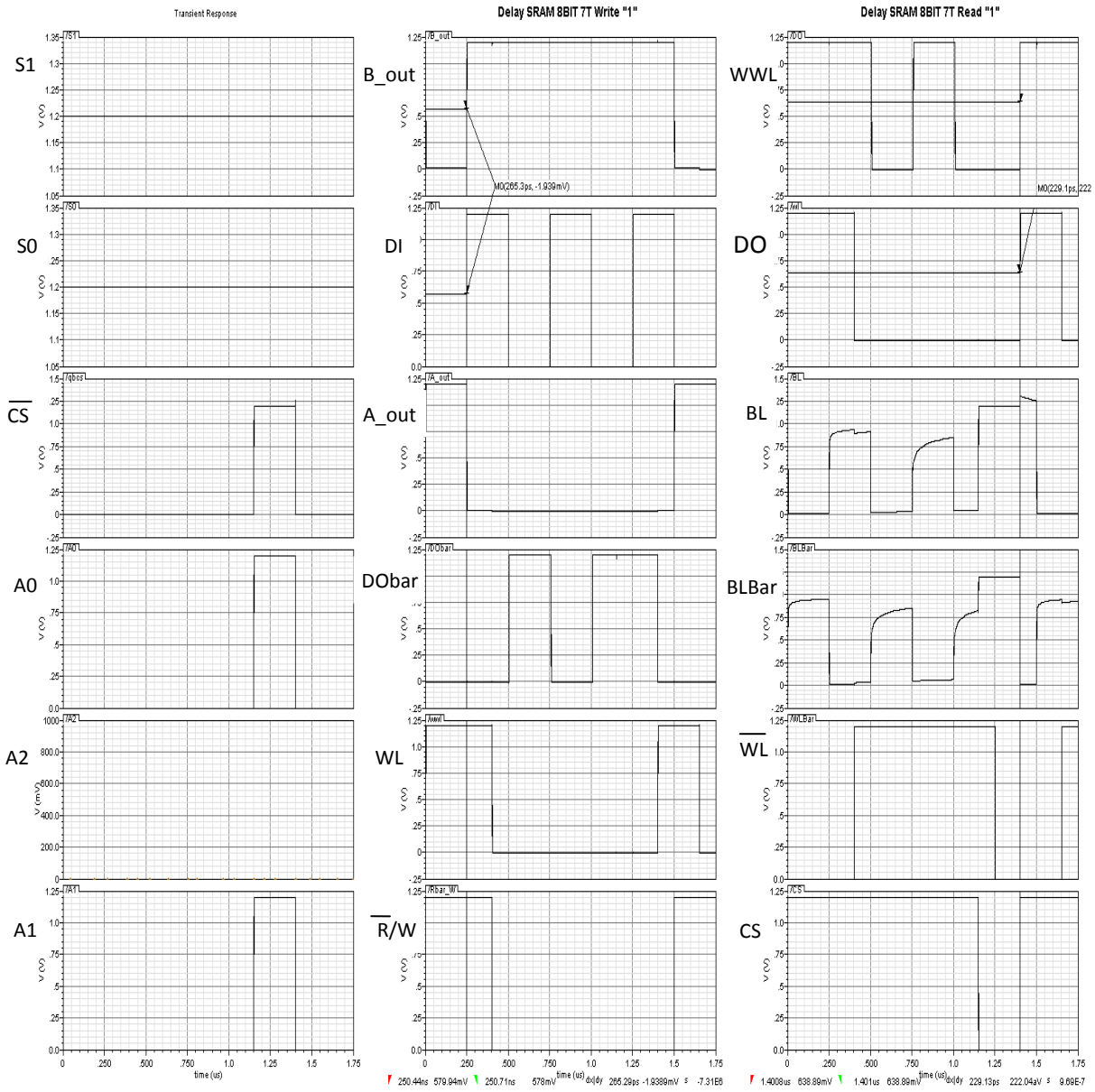

Figure 3.5 Waveforms showing write '1' and read '1' operaton of 7T SRAM 8BIT Cell.

3.3 Statistical Data Analysis

Monte Carlo simulation is a method for iteratively evaluating a system which is expressed in a closed loop equation [15]. The goal is to determine how random variation affects the sensitivity, performance or reliability of the system. Statistical Data Analysis is conducted on both 6T and 7T 8-BIT SRAMs. In Tables 3.3 and 3.4 the comparison is made on write and read operation between 6T and 7T 8-bit SRAM.

Table 3.3 Comparison between statistical analysis of SRAM 8-BIT 6T and 7T Write operation

SRAM 8BIT	Write operation	Mean(μ) (ps)	SD(δ) (ps)	δ /μ (%)
SRAM 6T	Write '0'	267.43	21.45	8.02
SRAM 7T		236.37	18.26	7.75
SRAM 6T	Write '1'	299.45	21.61	7.21
SRAM 7T		279.48	17.54	6.27

Table 3.4 Comparison between statistical analysis of SRAM 8-BIT 6T and 7T Write operation

SRAM 8BIT	Read operation	Mean (μ) (ps)	SD (δ) (ps)	δ /μ (%)
SRAM 6T	Read '0'	352.36	20.14	5.71
SRAM 7T		319.9	19.73	6.16
SRAM 6T	Read '1'	391.4	20.72	5.29
SRAM 7T		247.45	18.95	7.65

The closed loop equation is the delay equation between input and output. The Mean (μ) describes the central tendency or the location of the distribution. The standard deviation SD (δ) describes the spread of the distribution. As we infer from the tabulation 3.2 that the variation in delay of 7T and 6T SRAMs is minimum and is better for 7T SRAM because of additional transistor and variation in design of row decoder as there are additional wordlines to control. The process variation and mismatch were also considered while running the Monte Carlo simulations.

The percentage decrease in delay calculated from Tables 3.3 and 3.4, for write '0' operation is 11% and for write '1' operation is 6%. The percentage decrease in delay for read '0' operation is 9%, similarly for read '1' operation is 36%. Figures 3.6-3.13 explains about the statistical data analysis of 8-bit SRAM using 6T and 7T cells.

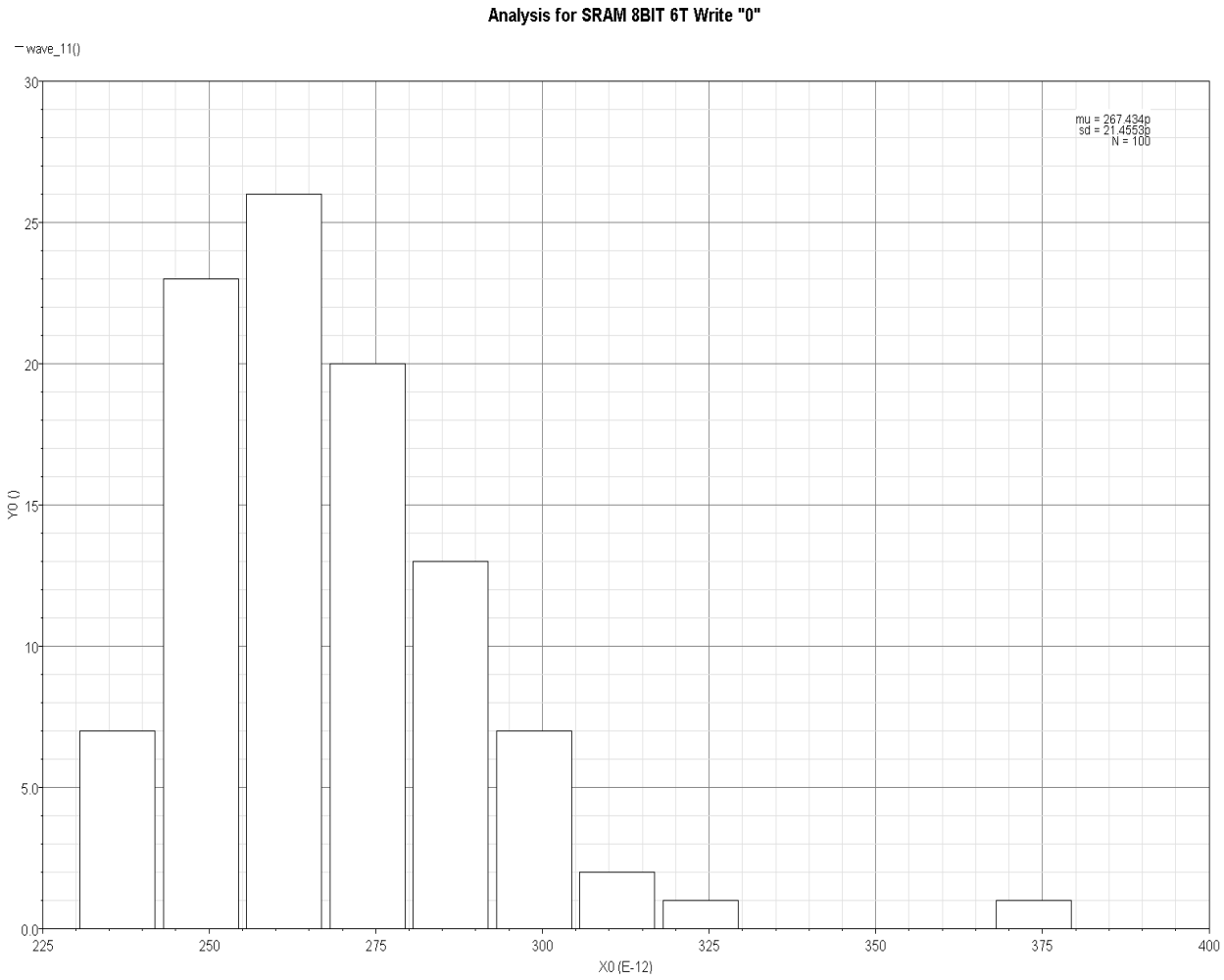

Figure 3.6 Monte Carlo simulation for 8-BIT SRAM 6T cell while performing write ‘0’ operation.

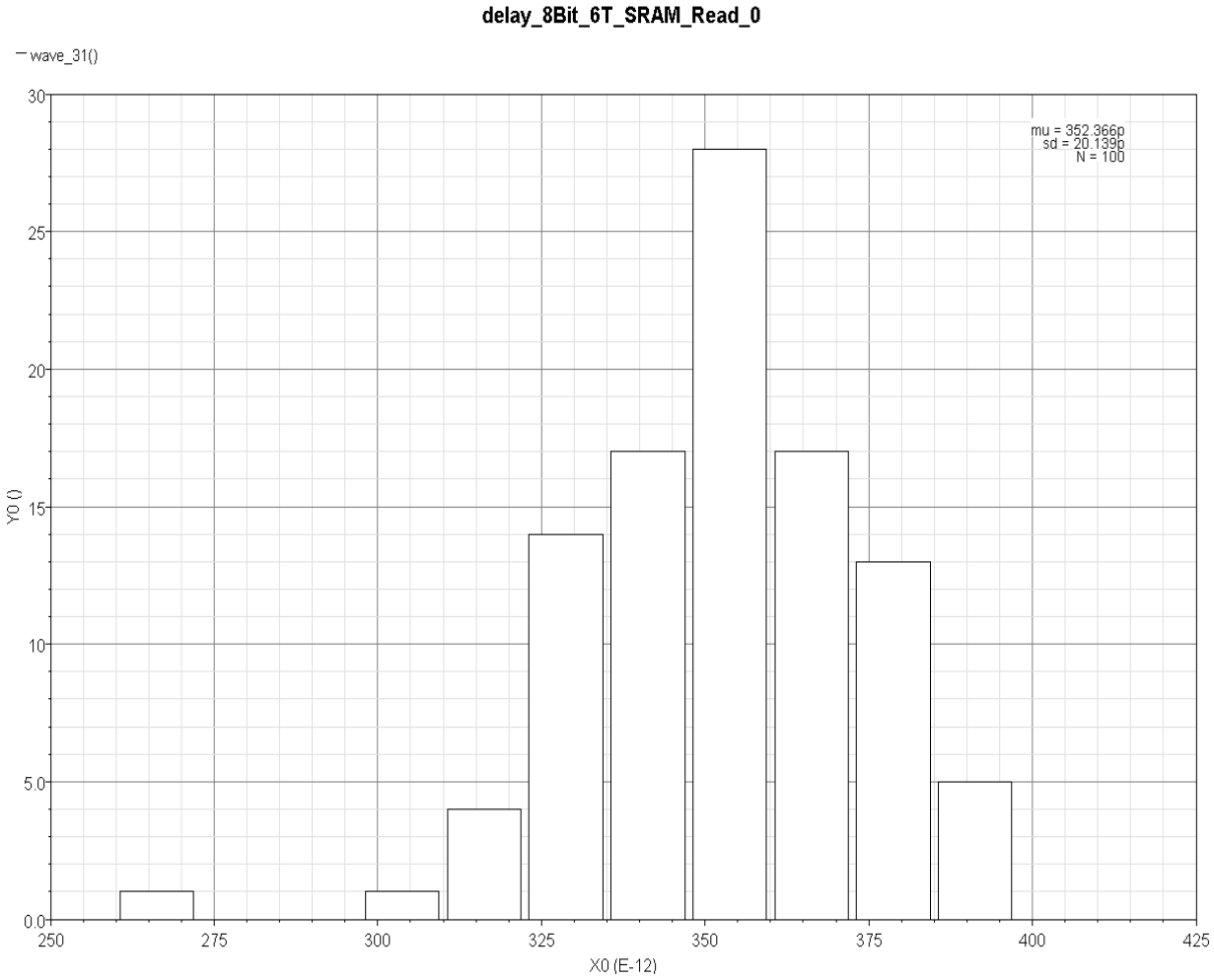

Figure 3.7 Monte Carlo Simulation for 8-Bit SRAM 6T while performing read ‘0’ operation.

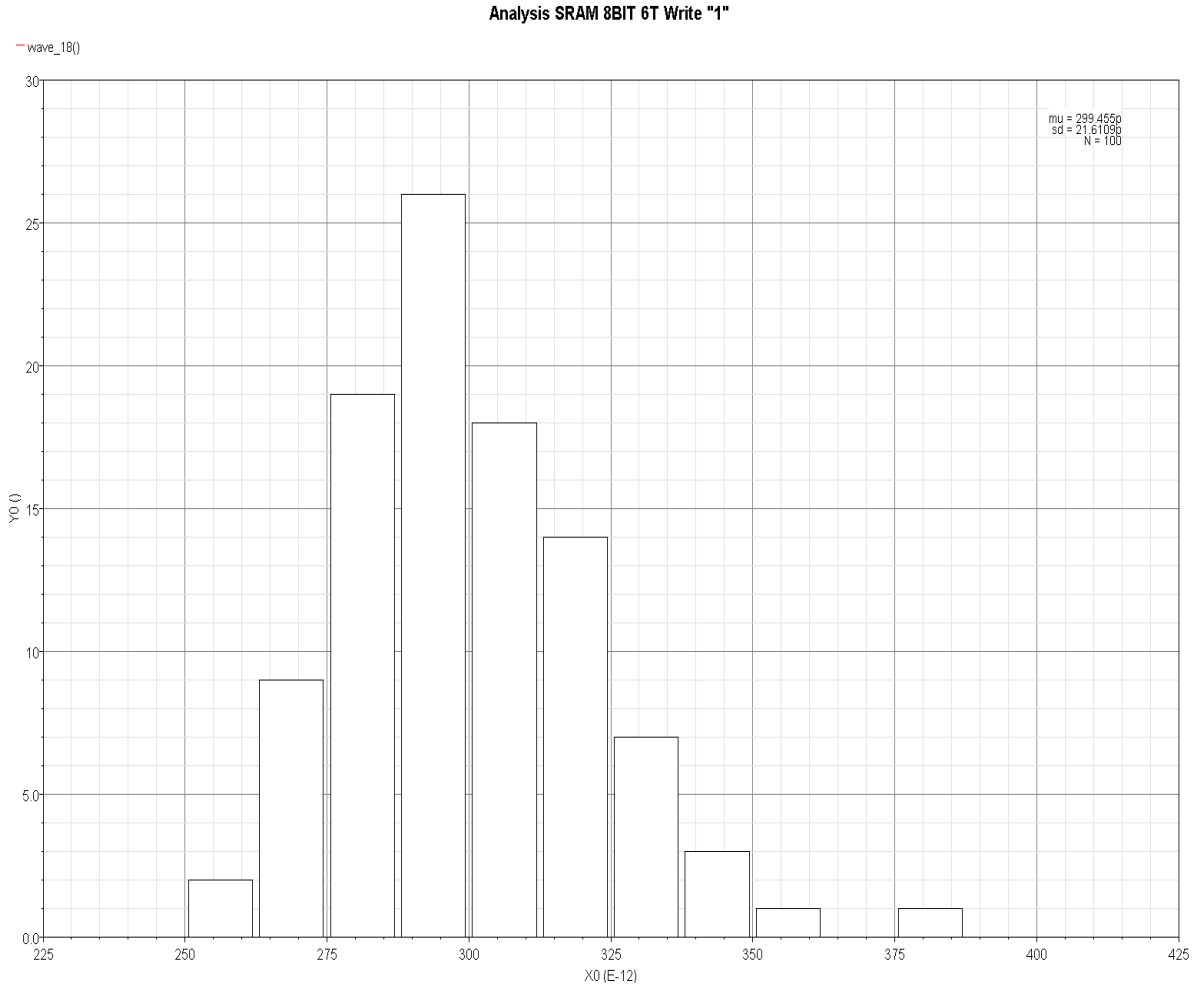

Figure 3.8 Monte Carlo Simulation for 8-Bit SRAM 6T while performing write ‘1’ operation.

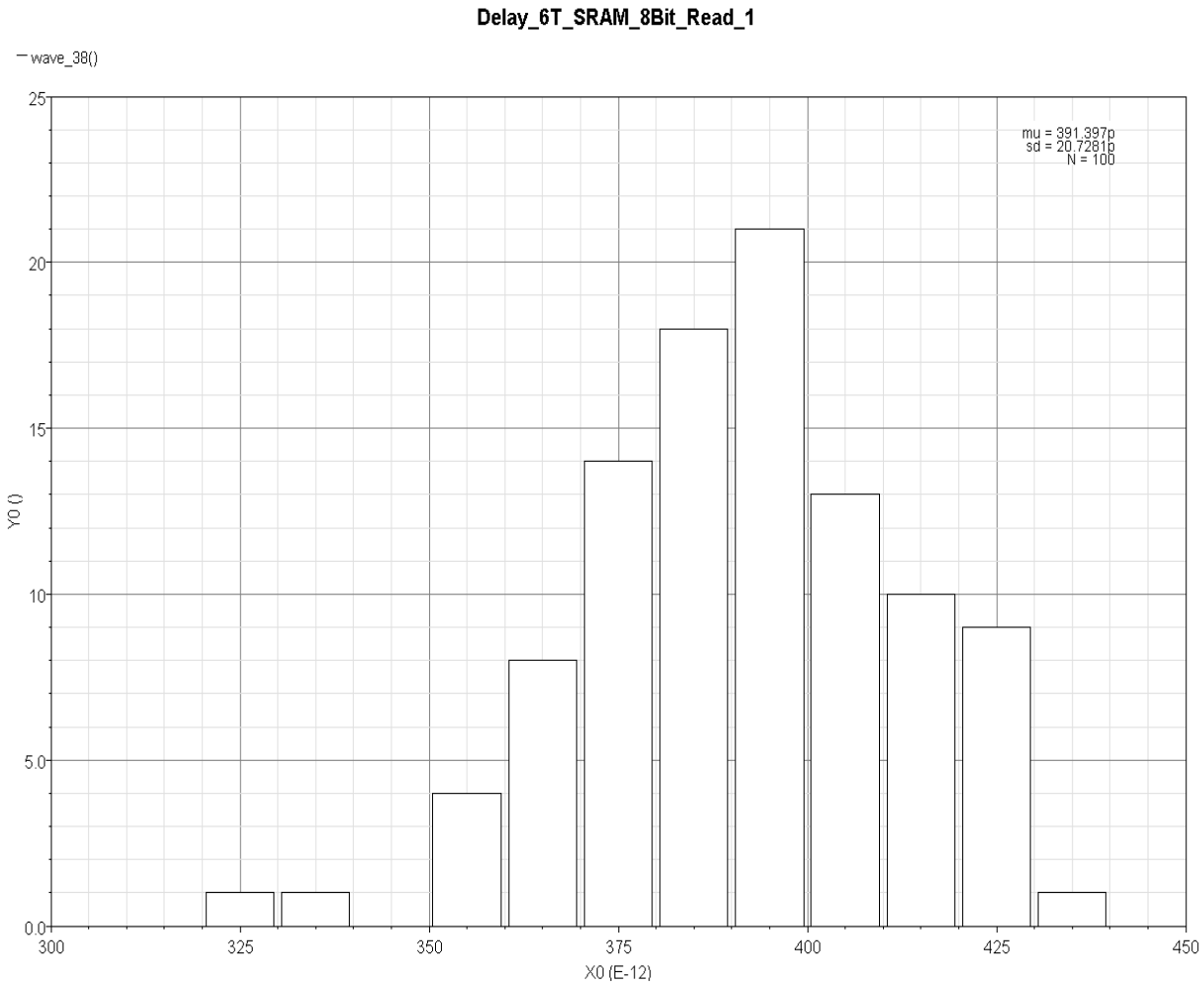

Figure 3.9 Monte Carlo Simulation for 8-Bit SRAM 6T while performing read '1' operation.

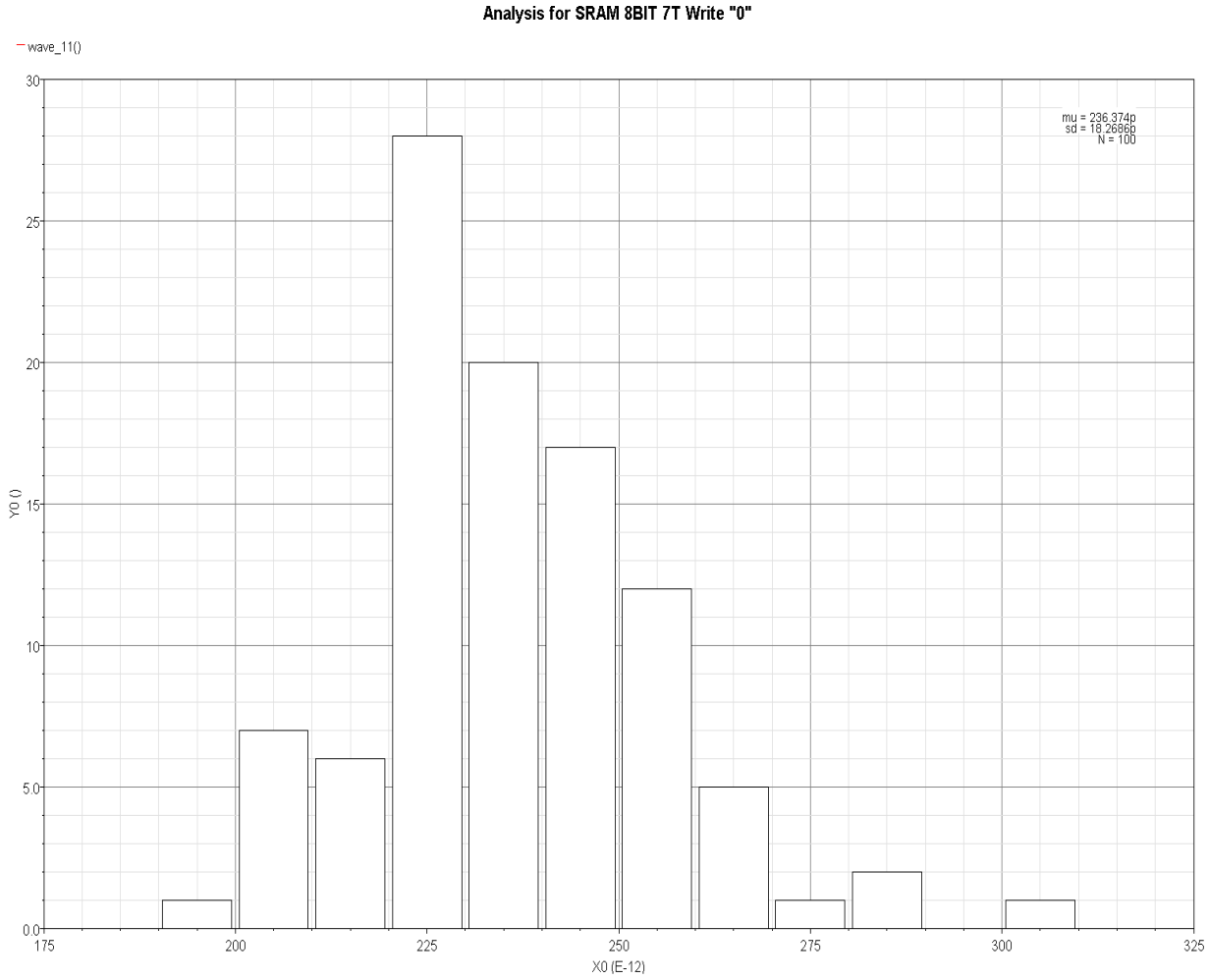

Figure 3.10 Monte Carlo Simulation for 8-Bit SRAM 7T while performing write '0' operation.

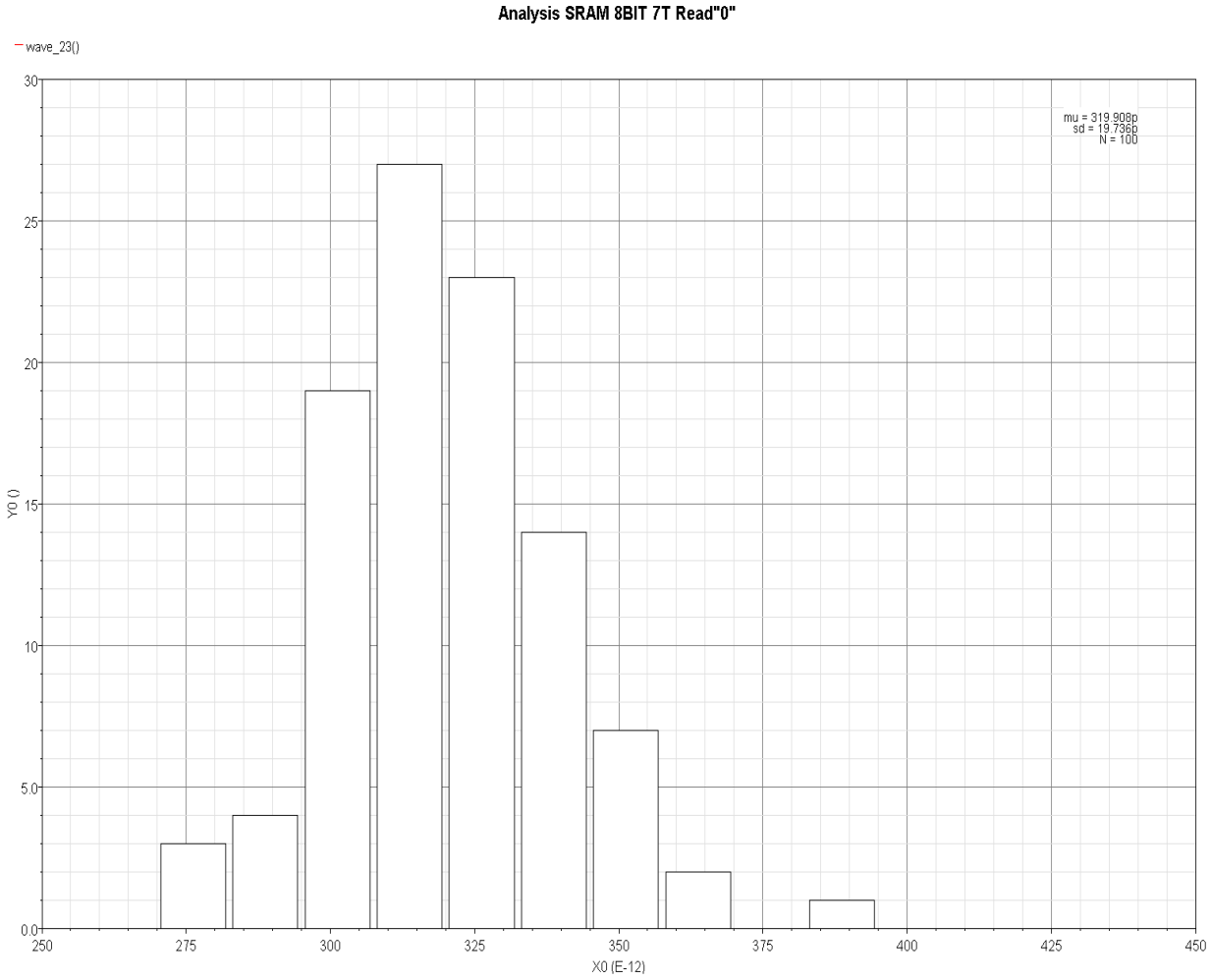

Figure 3.11 Monte Carlo Simulation for 8-Bit SRAM 7T while performing read ‘0’ operation.

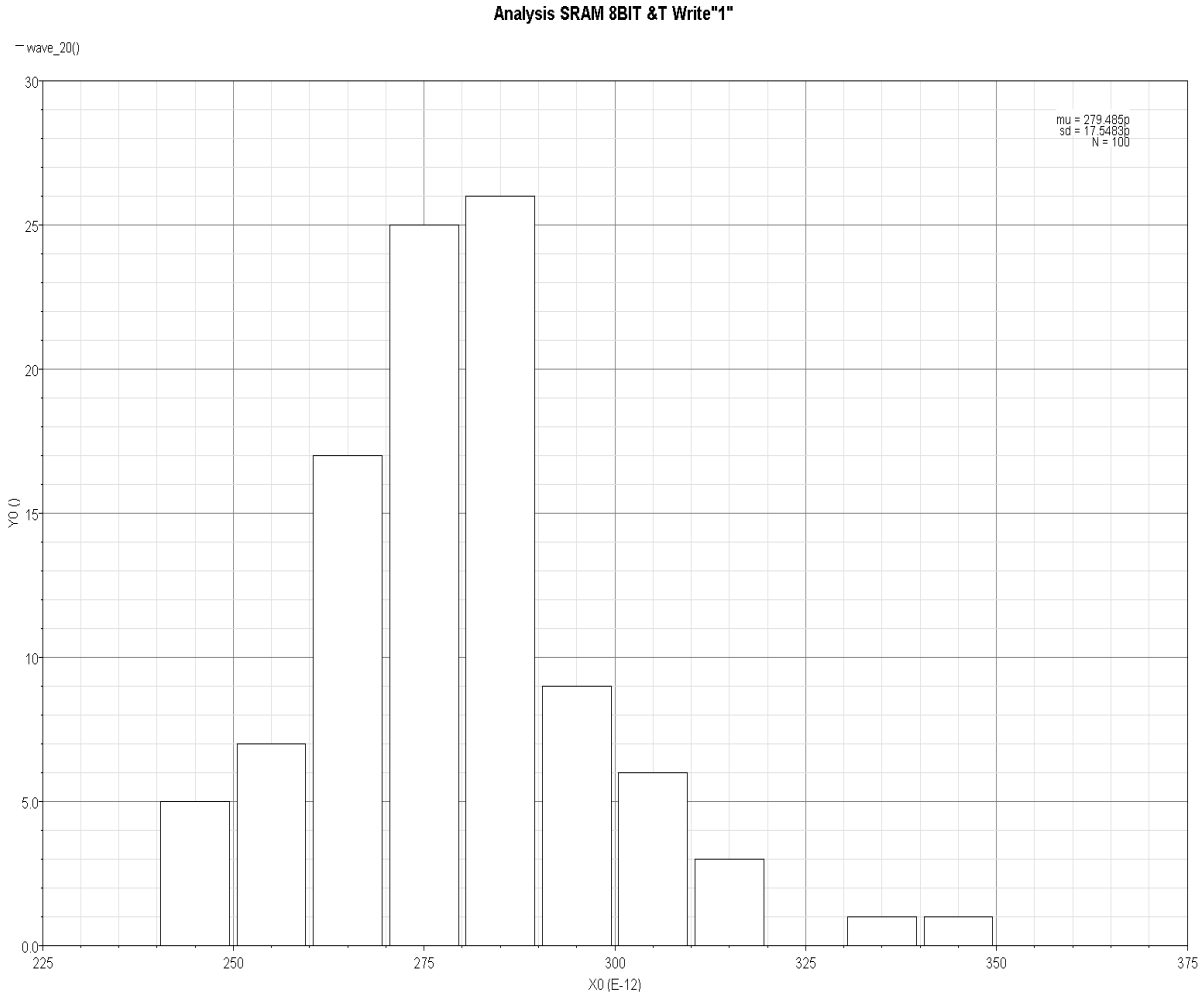

Figure 3.12 Monte Carlo Simulation for 8-Bit SRAM 7T while performing write ‘1’ operation.

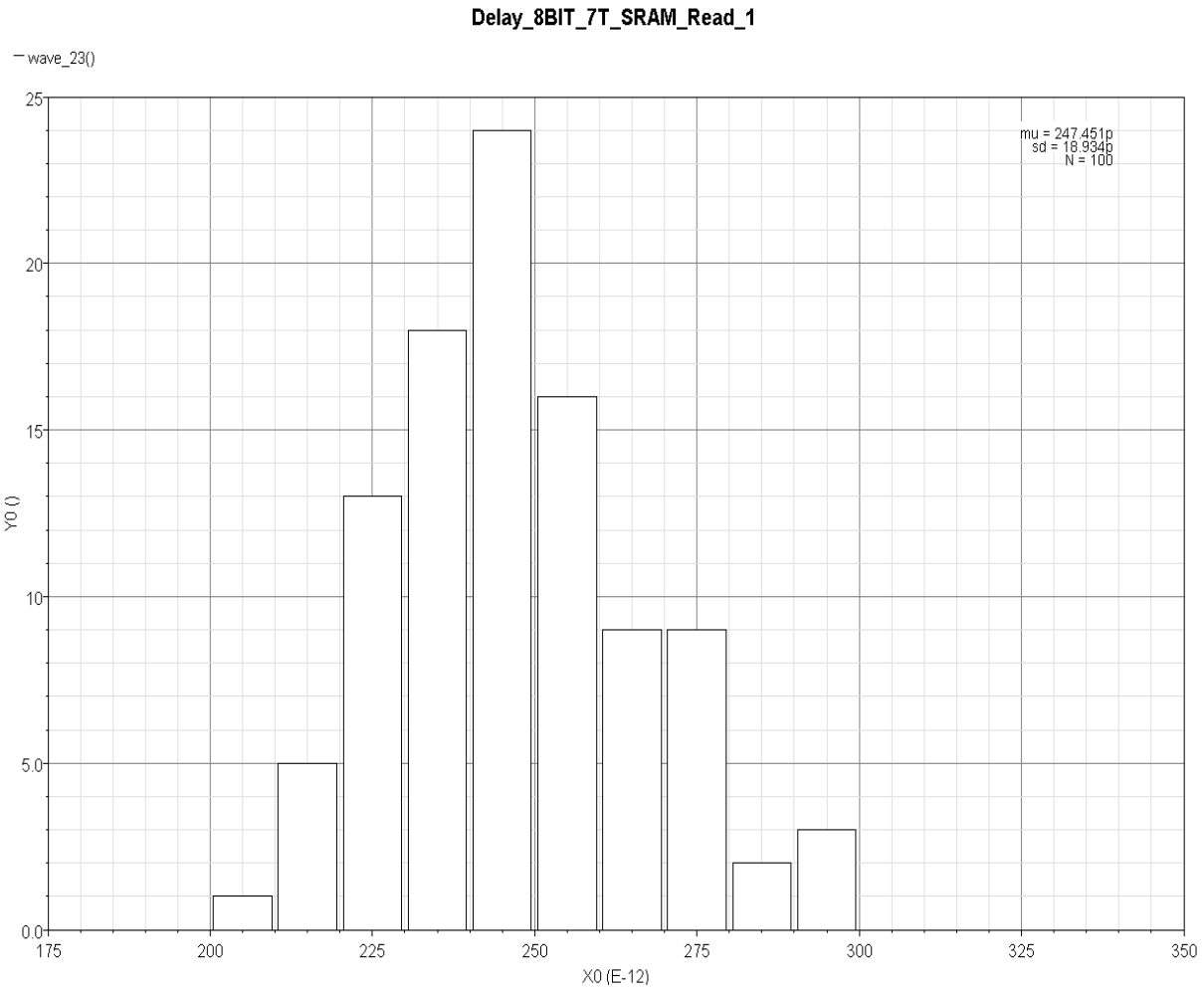

Figure 3.13 Monte Carlo Simulation for 8-Bit SRAM 7T while performing read ‘1’ operation.

3.4 Conclusion

The architecture and circuit implementation of SRAMs vary widely with application. The SRAM which is designed in this thesis work is best suited for “onchip RAM” applications such as scratch pad memory for microprocessor. The performance of this SRAM is fast as it is a static I/O circuit. The size of this RAM is limited because of architecture.

The comparison results show that the static noise margin (SNM) of the 7T SRAM cell is better than that of the 6T SRAM cell. The stability of the SRAM at low-V_{DD} is also proved by testing the SRAM at 720 mV. The average percentage decrease in delay for write operation from 6T 8-bit SRAM to 7T 8-bit SRAM design is 6% and the average percentage decrease in delay for read operation is 20%. The statistical analysis provides that variation in delay of the 7T SRAM is minimum for 100 simulations.

References

[1] EVERT SEEVINCK, Senior Member, IEEE, FRANS J. LIST, AND JAN LOHSTROH, Member, IEEE, "Static-noise margin analysis of MOS SRAM cells," *IEEE J. Solid-State Circuits*, vol. SC-22, no. 2, pp. 748–754, May 1987.

[2] Koichi Takeda, Yasuhiko Hagihara, Yoshiharu Aimoto, Masahiro Nomura, *Member, IEEE*, Yoetsu Nakazawa, Toshio Ishii, and Hiroyuki Kobatake, "A read-static-noise-margin-free SRAM cell for low-VDD and high-speed applications," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2005, pp. 478–479.

[3] The International Technology Roadmap for Semiconductors, 2004.

[4] P. Bai, C. Auth, S. Balakrishnan, M. Bost, R. Brain, V. Chikarmane, R. Heussner, M. Hussein, J. Hwang, D. Ingerly, R. James, J. Jeong, C. Kenyon, E. Lee, S.-H. Lee, N. Lindert, M. Liu, Z. Ma, T. Marieb, A. Murthy, R. Nagisetty, S. Natarajan, J. Neirynck, A. Ott, C. Parker, J. Sebastian, R. Shaheed, S. Sivakumar, J. Steigerwald, S. Tyagi, C. Weber, B. Woolery, A. Yeoh, K. Zhang, and M. Bohr "A 65 nm logic technology featuring 35 nm gate lengths, enhanced channel strain, 8 Cu interconnect layers, low-k ILD and 0.57 μm2 SRAM cell," in *IEDM Tech. Dig*, 2004, pp. 657–660.

[5] A. Chatterjee, J. Yoon, S. Zhao, S. Tang, K. Sadra, S. Crank, H. Mogul, R. Aggarwal, B. Chatterjee, S. Lytle, C. T. Lin, K. D. Lee, J. Kim, Q. Z. Hong, T. Kim, L. Olsen, M.Quevedo-Lopez, K. Kirmse, G. Zhang, C. Meek, D. Aldrich, H. Mair, M. Mehrotra, L. Adam, D. Mosher, J. Y. Yang, D. Crenshaw, B. Williams, J. Jacobs, M. Jain, J. Rosal, T.Houston, J. Wu, N. S. Nagaraj, D. Scott, S. Ashburn,

and A. Tsao. “A 65 nm CMOS technology for mobile and digital signal processing applications,” in *IEDM Tech. Dig., 2004*, pp. 665–668.

[6] Z. Luo, A. Steegen, M. Eller, M. Mann, C. Baiocco, P. Nguyen, L. Kim, M. Hoinkis, V. Ku, V. Klee, F. Jamin, P. Wrschka, P. Shafer, W. Lin, S. Fang, A. Ajmera, W. Tan, D. Park, R. Mo, J. Lian, D. Vietzke, C. Coppock, A. Vayshenker, T. Hook, V. Chan, K. Kim, A. Cowley, S. Kim, E. Kaltalioglu, B. Zhang, S. Marokkey, Y. Lin, K. Lee, H. Zhu, M. Weybright, R. Rengarajan, J. Ku, T. Schiml, J. Sudijono, I. Yang, and C. Wann, “High performance and low power transistors integrated in 65 nm bulk CMOS technology,” in *IEDM Tech. Dig, 2004*, pp. 661–664.

[7] E. Morifuji, T. Yoshida, H. Tsuno, Y. Kikuchi, S. Matsuda, S. Yamada, T. Noguchi, and M. Kakumu, “New guideline of Vdd and Vth scaling for 65 nm technology and beyond,” in *Proc. Symp. VLSI Dig. Tech. Papers*, 2004, pp. 164–165.

[8] Bharadwaj S. Amrutur and Mark A. Horowitz, “Speed and Power Scaling of SRAM’s”, *IEEE Transactions on Solid-State Circuits*, Vol. 35, No. 2, February 2000.

[9] Jean-Marc Masgonty1, Stefan Cserveny1, Christian Piguet, “Low-Power SRAM and ROM Memories” *CSEM, Neuchâtel, Switzerland and LAP-EPFL Lausanne, Switzerland.*

[10] Low Power VLSI design: an overview, third Edition by Abdellatif Bellaouar, Mohamed Elmasry.

[11] Digital MOS Integrated circuits by Niantsu Wang Design for Applications by Naintsu Wang, Prentice Hall, Edition 1989.

[12] Evelyn Grossar, Michele Stucchi, Karen Maex, Member, IEEE, and Wim Dehaene, Senior Member, "Read stability and write-ability analysis of SRAM Cells for nanometer Technologies" *IEEE Journal Of Solid-State Circuits*, Vol. 41, No. 11, November 2006.

[13] Benton H. Calhoun and Anantha P. Chandrakasan, "Static Noise Margin Variation for Sub-threshold SRAM in 65-nm CMOS", *IEEE Journal of Solid-State Circuits*, Vol. 41, No. 7, July 2006.

[14] Review of 6T SRAM cell by Ding – Ming Kwai Intellectual Property Library company June 3, 2005.

[15] http://www.vertex42.com/ExcelArticles/mc/MonteCarloSimulation.html

[16] Digital Integrated Circuits; A Design Perspective, Second Edition by Jan M. Rabaey, Anantha Chandrakasan and Borivoje Nikolic.

[17] http://en.wikipedia.com

[18] Eiji Morifuji, *Member, IEEE,* Dinesh Patil, *Student Member, IEEE,* Mark Horowitz, *Fellow, IEEE,* and Yoshio Nishi, *Fellow, IEEE* , "Power Optimization for SRAM and its scaling", *IEEE Transactions On Electron Devices,* VOL. 54, NO. 4, APRIL 2007.

[19] Nobutaro Shibata, Mayumi Watanabe, and Yasuyuki Tanabe, "A Current-Sensed High-Speed and Low-Power First-In–First-Out Memory Using a Wordline/Bitline-

Swapped Dual-Port SRAM Cell", *IEEE Journal Of Solid-State Circuits*, vol. 37, no. 6, June 2002.

[20] Koji Nii, *Member, IEEE*, Yasumasa Tsukamoto, Tomoaki Yoshizawa, Susumu Imaoka, Yoshinobu Yamagami, Toshikazu Suzuki, Akinori Shibayama, Hiroshi Makino, and Shuhei Iwade, "A 90-nm Low-Power 32-kB Embedded SRAM With Gate Leakage Suppression Circuit for Mobile Applications", *IEEE Journal Of Solid-State Circuits*, Vol. 39, No. 4, April 2004.

[21] K. Itoh, Senior Member, IEEE, Katsuro Sasaki, Member, IEEE, and Yoshinobu Nakagome, Member, IEEE, "Trends in Low-Power RAM Circuit Technologies", *Proc. IEEE*, Vol. 83, No 4, April 1995, pp. 524-543.

[22] N. Tzartzanis, W.C. Athas, and L. Svensson, "A Low-Power SRAM with Resonantly Powered Data, Address, Word and Bit Lines", *Proc. ESSCIRC'2000*, Sept. 19-21, 2000, Stockholm, pp. 336-339